



IBM DB2 Cloning Tool for z/OS User's Guide

Version 3 Release 1



IBM DB2 Cloning Tool for z/OS User's Guide

Version 3 Release 1

Note:

Before using this information and the product it supports, read the "Notices" topic at the end of this information.

Fourth Edition (February 2015)

This edition applies to Version 3 Release 1 of DB2 Cloning Tool for z/OS (product number 5655-N15) and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SC19-3493-02.

© Copyright IBM Corporation 1996, 2015; Copyright Rocket Software Inc. 1996, 2015. All Rights Reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About this information	ix	Displaying customization jobs	68
Chapter 1. DB2 Cloning Tool overview	1	Maintaining customization jobs.	68
What's new in DB2 Cloning Tool	1	Using Tools Customizer in a multiple-LPAR environment	69
What does DB2 Cloning Tool do?	10	Chapter 5. After customizing DB2 Cloning Tool using Tools Customizer	71
Cloning definitions	11	Reviewing the CKZINI PARMLIB member variables	71
Why clone a DB2 subsystem?	11	Verifying successful customization.	72
Can I clone a DB2 subsystem without DB2 Cloning Tool?.	12	Adding DB2 Cloning Tool to the DB2 Administration Tool Launchpad	72
DB2 support	12	DB2 version migration considerations	72
Volume copy products supported by DB2 Cloning Tool	12	Chapter 6. Planning for copying and renaming volumes.	75
DB2 Cloning Tool features and benefits	13	Selection of source and target volumes	75
Service updates and support information	14	Volume relationship conflicts	78
Product documentation and updates	14	Ensure RACF and ACF authorities.	78
Accessibility features	15	Data set renaming considerations	78
Chapter 2. Preparing to customize DB2 Cloning Tool	17	Target ICF catalog considerations	78
Set up your environment prior to customization	18	Location of the source and target ICF catalogs.	79
Function authorization requirements	20	Target data set ICF catalog aliases	80
Use of the U.S.A. EBCDIC code set	22	Return code choices	80
Filtering pattern masks	23	Cloning a DB2 subsystem	80
Worksheets: Gathering required data set names	25	Chapter 7. Setting up to copy by volume with FlashCopy, SnapShot, or TimeFinder/Clone	83
APF authorizing load libraries	27	DB2 Cloning Tool process for volume FlashCopy, SnapShot, or TimeFinder/Clone	83
Worksheets: Gathering parameter values for Tools Customizer	28	Steps for volume FlashCopy, SnapShot, or TimeFinder/Clone	84
Chapter 3. Starting and preparing Tools Customizer for use	45	Chapter 8. Setting up to copy by volume with an onsite mirror tool	87
Starting Tools Customizer	45	DB2 Cloning Tool process for user-created target volume clones	87
Modifying Tools Customizer user settings	46	Steps for user-created target volumes.	88
Chapter 4. Customizing DB2 Cloning Tool	51	DB2 Cloning Tool process for EMC TimeFinder/Mirror target volume clones	90
Roadmap: Customizing DB2 Cloning Tool for the first time	51	Steps for EMC TimeFinder/Mirror target volume clones	90
Roadmap: Customizing a new version of DB2 Cloning Tool from a previous customization	52	Chapter 9. Cloning DB2 subsystems	95
Roadmap: Recustomizing DB2 Cloning Tool	53	DB2 offline cloning.	95
Specifying the metadata library for the product to customize	54	DB2 offline cloning procedure	96
Discovering DB2 Cloning Tool information automatically.	56	Steps for DB2 offline cloning	96
Creating and associating DB2 entries	58	DB2 offline cloning with removal of data sharing members procedure	98
Defining parameters	59	Steps for DB2 offline cloning with removal of data sharing members.	99
Defining DB2 Cloning Tool parameters	60	DB2 offline cloning with target becoming non-data sharing procedure	101
Defining DB2 parameters.	61		
Generating customization jobs	63		
Submitting customization jobs	63		
Browsing parameters	65		
Copying DB2 entries	66		
Removing DB2 entries.	67		
Deleting DB2 entries	68		

Steps for DB2 offline cloning with target becoming non-data sharing	102
DB2 online cloning	104
Status of transactions in flight	104
DB2 online cloning procedure	105
Steps for DB2 online cloning	107
DB2 online cloning with removal of data sharing members procedure	111
Steps for DB2 online cloning with removal of data sharing members	113
DB2 online cloning with target becoming non-data sharing procedure	117
Steps for DB2 online cloning with target becoming non-data sharing	118

Chapter 10. Subsystem cloning using the DB2 Cloning Tool stored procedure 123

About the DB2 Cloning Tool stored procedure	123
Steps for cloning a DB2 subsystem using the stored procedure	124
Parameter files and parameter descriptions	124
Calling the stored procedure	135
Verifying the cloning	137
JCL members generated by the stored procedure	137
Running the verified cloning	138
Recloning a DB2 system	138
Stored procedure example: Cloning a non data sharing subsystem.	139
Stored procedure example: Cloning from a system-level backup	140
Stored procedure example: Cloning from DB2 system-level backup dump tapes	141
Stored procedure example: Cloning from DB2 system-level backup dump tapes across sysplexes	143

Chapter 11. DB2 Cloning Tool Table Space Cloning overview. 147

What does DB2 Cloning Tool Table Space Cloning do?	147
Why you should use DB2 Cloning Tool Table Space Cloning	148
DB2 Cloning Tool Table Space Cloning features and benefits	149
DB2 Cloning Tool Table Space Cloning terminology - DB2 objects vs. VSAM objects	149
Use of the U.S.A. EBCDIC code set	150

Chapter 12. Planning for copying DB2 table spaces by data set 151

Considerations for in-progress read/write activity or DB2 utilities	151
Considerations for the DB2 Cloning Tool Table Space Cloning cloning process.	151
Subsystem copy rules	151
Renaming table spaces and index spaces	152
Identity columns	152
LOBs, BLOBs, CLOBs, and DBCLOBs	152
XML considerations	152

DDF and TCP/IP considerations	153
Object attributes	153
APF authorization	154
Ensure RACF and ACF authorities	154
Missing target table spaces and index spaces	155
Considerations for target objects created using DEFINE NO.	155
Non-SMS managed volumes	156
Space processing order	156
How DB2 Cloning Tool Table Space Cloning starts and stops table spaces and index spaces	156
Message output	158
Considerations for DB2 Version 9.1 and later	158
Considerations for data set renaming	159
Considerations for using FUZZY-COPY	160
Considerations for generating target object DDL using PROCESS-DDL.	160
Sample DDL processing scenarios	161
Considerations for using catalog prefetch to populate the object cache	163
Considerations for caching indexes	164
Using a local TCP/IP server to populate the target cache for multiple source copy jobs	165
Return code choices	166
Plans and packages	166
Dropped and altered tables.	166
Copying versioned objects	166

Chapter 13. DB2 Cloning Tool Table Space Cloning function overview. 169

Source job overview	169
Target job overview	169
TCP/IP job overview (optional)	170
Report job overview (optional)	170
The source job	170
Source job - Init phase I	170
Source job - Discovery phase II	171
Source job - Eligibility phase III	171
Source job - Copy phase IV.	172
The target job	173
The TCP/IP server job (optional).	174
The report job (optional).	175
CKZFIX job	178

Chapter 14. Setup procedures for copy by data set with FlashCopy, SnapShot, or TimeFinder/Clone 179

Setup for scenario 1: FlashCopy, SnapShot, or TimeFinder/Clone - All target table spaces and index spaces already exist in DB2 catalog	179
Step 1: Create target DB2 table spaces and index spaces in the target DB2 catalog	179
Step 2: Set up and execute source job (CKZ1SRC)	180
Step 3: Set up and execute target job (CKZTRG)	184
Setup for scenario 2: FlashCopy, SnapShot, or TimeFinder/Clone - Some or all target table spaces and index spaces do not exist in DB2 catalog.	187
Step 1: Create target DB2 table spaces and index spaces in the target DB2 catalog (if required)	188

Step 2: Set up and execute source job (CKZ2SRC)	189
Step 3 (Optional): Create missing target table spaces and index spaces.	193
Step 4: Set up and execute optional FIX job (CKZFIX).	194
Step 5: Set up for a re-run of the source job	195
Step 6: Set up and execute target job (CKZTRG)	195

Chapter 15. Set up procedures for copy by data set for all other methodologies 199

Setup for scenario 3: Other copy methodologies - all target table spaces and index spaces already exist in DB2 catalog	199
Step 1: Verify/create target DB2 table spaces and index spaces in target DB2 catalog	199
Step 2: Set up and execute source job (CKZ3SRC)	200
Step 3 (Optional): Set up and execute stop target job (CKZSTPT)	204
Step 4 (Optional): Set up and execute stop source job (CKZSTPS)	205
Step 5: Set up and execute non-DB2 Cloning Tool Table Space Cloning job to execute data set copies outside DB2 Cloning Tool Table Space Cloning	206
Step 6 (Optional): Set up and execute job to re-start source DB2 table spaces and index spaces (CKZSTRS).	206
Step 7: Set up and execute target job (CKZTRG)	207
Setup for scenario 4: Other copy methodologies - some or all target table spaces and index spaces do not exist in DB2 catalog	211
Step 1: Create target DB2 table spaces and index spaces in target DB2 catalog	211
Step 2: Set up source job (CKZ4SRC)	212
Step 3 (Optional): Set up stop target job (CKZSTPT)	217
Step 4 (Optional): Set up stop source job (CKZSTPS)	218
Step 5: Set up non-DB2 Cloning Tool Table Space Cloning job to execute data set copies outside DB2 Cloning Tool Table Space Cloning	219
Step 6 (Optional): Set up job to re-start source DB2 table spaces and index spaces (CKZSTRS)	219
Step 7 (Optional): Create missing target table spaces and index spaces.	220
Step 8 (Optional): Set up job to make copied VSAM objects available to target DB2 (CKZFIX).	221
Step 9: Set up for a re-run of the source job	222
Step 10: Set up and execute target job (CKZTRG)	222

Chapter 16. Procedures for cloning DB2 table spaces and index spaces 227

Copy procedure 1: FlashCopy, SnapShot, or TimeFinder/Clone setup - when all target table spaces and index spaces already exist in DB2 catalog	227
---	-----

Step 1: Verify/create target DB2 table spaces and index spaces in target DB2 catalog	227
Step 2 (Optional): Submit TCP/IP server job	227
Step 3: Submit source job	228
Step 4: Submit target job	228
Copy procedure 2: FlashCopy, SnapShot, or TimeFinder/Clone setup - when some or all target table spaces and index spaces do not exist in DB2 catalog	228
Step 1: Verify/create target DB2 table spaces and index spaces in target DB2 catalog	229
Step 2 (Optional): Submit TCP/IP server job	229
Step 3: Submit source job	229
Step 4: Create missing target DB2 table spaces and index spaces - manually or using non-DB2 Cloning Tool Table Space Cloning Tool	230
Step 5: Submit FIX job on target z/OS system (CKZFIX).	230
Step 6: Rerun source job.	231
Step 7: Submit target job	231
Copy procedure 3: All other copy methodologies - All target table spaces and index spaces already exist in DB2 catalog	231
Step 1: Verify/create target DB2 table spaces and index spaces in target DB2 catalog	232
Step 2 (Optional): Submit TCP/IP server job	232
Step 3: Submit source job	232
Step 4: Submit job CKZSTPT to stop target DB2 table spaces and index spaces	233
Step 5: Submit job CKZSTPS to stop source DB2 table spaces and index spaces	233
Step 6: Submit non-DB2 Cloning Tool Table Space Cloning job to copy VSAM objects	233
Step 7: Submit job CKZSTRS to start source DB2 table spaces and index spaces	233
Step 8: Rerun source job.	233
Step 9: Submit target job	233
Copy procedure 4: All other copy methodologies - some or all target table spaces and index spaces do not exist in DB2 catalog	233
Step 1: Verify/create target DB2 table spaces and index spaces in target DB2 catalog	234
Step 2 (Optional): Submit TCP/IP server job	234
Step 3: Submit source job	235
Step 4: Create missing target DB2 table spaces and index spaces	236
Step 5: Submit job CKZSTPT to stop target DB2 table spaces and index spaces	236
Step 6: Submit job CKZSTPS to stop source DB2 table spaces and index spaces	236
Step 7: Submit non-DB2 Cloning Tool Table Space Cloning job to copy VSAM objects	236
Step 8: Submit job CKZSTRS to restart source DB2 table spaces and index spaces	236
Step 9: Submit FIX job on target z/OS system (CKZFIX).	236
Step 10: Rerun the source job to prepare target job	237
Step 11: Submit target job	237

Chapter 17. Using data masking with table space cloning	239
Source job changes	240
Updating the template for the job to rebuild indexes	241
Specifying the masking rule	241
MASKRULE(STATIC, <i>numeric</i> <i>"date_or_time"</i> <i>"string"</i>)	241
MASKRULE(MASK, <i>"pattern"</i> , <i>start</i> , <i>end</i>)	242
MASKRULE(PATTERN, <i>"pattern"</i> , <i>"use_sources"</i>)	244
MASKRULE(RANDOM, <i>"min"</i> , <i>"max"</i> , <i>"check_bounds"</i>)	246
MASKRULE(RANDOM, <i>"min_timestampz"</i> , <i>"max_timestampz"</i> , <i>"min_timezone"</i> , <i>"max_timezone"</i> , <i>"check_bounds"</i>)	247
MASKRULE(SEQUENCE, <i>initial_value</i> , <i>increment</i>)	248
MASKRULE(SCRAMBLE)	248
MASKRULE(CURRENT DATE)	
MASKRULE(CURRENT TIME)	
MASKRULE(CURRENT TIMESTAMP, <i>"timezone"</i>)	249
MASKRULE(CURRENT USER)	249
Target job changes	250
Submitting the job to rebuild indexes	251
Examples of masking jobs	251
Chapter 18. Using image copies to clone table spaces and index spaces	259
Steps for cloning from image copies (including index rebuilds)	260
Selecting a specific image copy to use as the cloning source	261
Source job changes for cloning from image copies	262
Options for rebuilding indexes when cloning table spaces and index spaces from image copies	262
Chapter 19. Using LOG-APPLY to make consistent copies of table spaces and index spaces	265
Source job changes	266
Source TCP/IP server job (required when using LOG-APPLY across multiple LPARs)	266
Target job changes	267
Options for rebuilding indexes	267
Method 1: Using job templates and the JOB-TEMPLATE command to rebuild indexes	268
Method 2: Using TARGET-JOB-INDEX-REBUILD-DDN to rebuild indexes	268
Chapter 20. Using job templates	269
Predefined processing variables	269
Data set processing variables	271
Block processing variables	271
Job templates for utilities	272
Chapter 21. Using the ISPF interface	275
The DB2 Cloning Tool ISPF interface	275

Configuring DB2 subsystems	276
Configuring a subsystem	276
Configuring user settings	278
Setting user defaults (job card and work data sets)	278
Setting subsystem cloning defaults	279
Setting table space cloning defaults	288
Creating cloning jobs using the interface	316
Creating a profile	317
Subsystem cloning	317
Subsystem cloning steps summary	318
Step summary for subsystem cloning from a system-level backup	318
Select source and target subsystems	319
Add or verify high level qualifiers	321
Specify source and target volume pairings	321
Specify source and target ICF catalogs	322
Specify rename masks for source and target data sets	322
Build the cloning jobs from a profile	322
Submit the jobs	327
Table space cloning	327
Table space cloning steps summary	328
Select the source and target DB2 subsystems	328
Edit the source job	329
Verify the target job settings	330
Verify the report job settings	330
Verify the TCP/IP server job settings	331
Verify the source TCP/IP server job settings	331
Build the table space cloning jobs from a profile	331
Submit the jobs	333
Chapter 22. Cloning table spaces from DB2 Administration Tool	335
Chapter 23. DB2 Cloning Tool Subsystem Cloning commands	337
BACKINFO-REFORMAT	338
BACKINFO-REFORMAT command syntax	339
BACKINFO-REFORMAT command and keyword definitions	340
BACKINFO-REFORMAT step JCL examples	342
BCSCLEAN	345
BCSCLEAN command syntax	346
BCSCLEAN command and keyword definitions	346
BCSCLEAN step JCL examples	349
COPY	350
COPY command syntax	351
COPY command and keyword definitions	353
COPY step JCL example	368
COPYCHECK	371
COPYCHECK command syntax	371
COPYCHECK command and keyword definitions	372
COPYCHECK step JCL example	373
DB2ALTERBSDS	374
DB2ALTERBSDS command syntax	374
DB2ALTERBSDS command and keyword definitions	375
DB2ALTERBSDS step JCL example	377

DB2FIX	379	JRNLUPTGRADE step JCL example	430
DB2FIX command syntax	381	ONLINECLIP	431
DB2FIX command and keyword definitions	381	ONLINECLIP command syntax	431
DB2FIX step JCL example	384	ONLINECLIP command and keyword	
DB2GETBACKINFO	385	definitions	431
DB2GETBACKINFO command syntax	385	ONLINECLIP step JCL example	432
DB2GETBACKINFO command and keyword		RENAME	433
definitions	386	RENAME command syntax	433
DB2GETBACKINFO step JCL examples	388	RENAME command and keyword definitions	436
DB2LGRNXCLEAN	390	RENAME step JCL example	448
DB2LGRNXCLEAN command syntax	391	RESTORE-FROM-DUMPTAPES	450
DB2LGRNXCLEAN command and keyword		RESTORE-FROM-DUMPTAPES command	
definitions	391	syntax	450
DB2LGRNXCLEAN step JCL example	392	RESTORE-FROM-DUMPTAPES command and	
DB2RBLDBSDS	393	keyword definitions	451
DB2RBLDBSDS command syntax	393	RESTORE-FROM-DUMPTAPES step JCL	
DB2RBLDBSDS command and keyword		example	454
definitions	393	UCATOPTIONS	456
DB2RBLDBSDS step JCL example	395	UCATOPTIONS command syntax	456
DB2SETLOG	396	UCATOPTIONS command and keyword	
DB2SETLOG command syntax	396	definitions	456
DB2SETLOG command and keyword definitions	397	UCATOPTIONS step JCL example	458
DB2SETLOG step JCL example	397	VARYOFF	459
DB2SQL	399	VARYOFF command syntax	459
DB2SQL command syntax	399	VARYOFF command and keyword definitions	459
DB2SQL command and keyword definitions	400	VARYOFF step JCL example	461
DB2SQL step JCL example	403	VARYON	463
DB2START	404	VARYON command syntax	464
DB2START command syntax	405	VARYON command and keyword definitions	464
DB2START command and keyword definitions	405	VARYON step JCL example	466
DB2START step JCL example	408	VOLOPTIONS	468
DB2STOP	409	VOLOPTIONS command syntax	469
DB2STOP command syntax	409	VOLOPTIONS command and keyword	
DB2STOP command and keyword definitions	409	definitions	469
DB2STOP step JCL example	410	VOLOPTIONS step JCL example	471
DB2UPDATE	411	Chapter 24. DB2 Cloning Tool Table	
DB2UPDATE command syntax	412	Space Cloning commands	473
DB2UPDATE command and keyword		COPY	473
definitions	413	COPY command syntax	474
DB2UPDATE step JCL example	417	COPY command and keyword definitions	476
DB2UTILXCLEAN	420	OBJECT-TRANSLATE considerations	504
DB2UTILXCLEAN command syntax	421	DDL-ATTRIBUTE-CHANGE parameter values	506
DB2UTILXCLEAN command and keyword		GLOBAL command values for EMC	
definitions	421	TimeFinder/Clone Mainframe Snap Facility data	
DB2UTILXCLEAN step JCL example	422	set level support	508
Step JCL example to identify objects with UT		DATASUBTYPE	510
status	423	DATASUBTYPE command syntax	510
DB2XCFCLEAN	424	DATASUBTYPE command & keyword	
DB2XCFCLEAN command syntax	424	definitions	511
DB2XCFCLEAN command and keyword		HLQDDDF	511
definitions	424	HLQDDDF command syntax	511
DB2XCFCLEAN step JCL example	425	HLQDDDF command and keyword definitions	512
FINDUCATS	426	LISTDEF	512
FINDUCATS command syntax	426	LISTDEF command syntax	513
FINDUCATS command and keyword		LISTDEF command and keyword definitions	514
definitions	426	SET	516
FINDUCATS step JCL example	427	SET command syntax	516
JRNLUPTGRADE	428	SET command and keyword definitions	517
JRNLUPTGRADE command syntax	429		
JRNLUPTGRADE command and keyword			
definitions	429		

**Chapter 25. Tools Customizer
reference 529**

Tools Customizer terminology and data sets 529
Tools Customizer terminology. 529
Data sets that Tools Customizer uses during
customization 531

Chapter 26. Reference 535

Cloning scenarios 535
Volume cloning using an interim set of volumes 535
DB2 subsystem cloning from a DB2 BACKUP
SYSTEM backup 538
DB2 subsystem cloning to a specific point in
time from a DB2 BACKUP SYSTEM backup 547
DB2 subsystem cloning from DB2 BACKUP
SYSTEM dump tapes. 559
DB2 subsystem cloning from other system level
backups when backup volumes are online. 566
DB2 subsystem cloning from other system level
backups when backup volumes are offline. 570

CKZINI customization values 575
Structure of the CKZINI member. 575
Syntax rules. 575
About DB2 Cloning Tool Table Space Cloning
token errors 577
CKZINI keyword syntax and descriptions. 577
Deallocating target DB2 coupling facility structures 591
Troubleshooting 592
Messages. 592
How to look up message explanations 931
Gathering diagnostic information. 931
Validating load module contents with
MODLEVEL. 932

Notices 935

Index 939

About this information

DB2® Cloning Tool for z/OS® (also referred to as DB2 Cloning Tool) is a query analysis tool that you can use to fine-tune complicated queries so that they run as efficiently as possible.

These topics provide instructions for installing, configuring, and using DB2 Cloning Tool.

These topics are designed to help database administrators, system programmers, application programmers, and system operators perform these tasks:

- Plan for the installation of DB2 Cloning Tool
- Install and operate DB2 Cloning Tool
- Customize your DB2 Cloning Tool environment
- Diagnose and recover from DB2 Cloning Tool problems
- Design and write applications for DB2 Cloning Tool
- Use DB2 Cloning Tool with other DB2 or IMS™ products

Always check the DB2 Tools Product Documentation page for the most current version of this information:

<http://www.ibm.com/software/data/db2imstools/db2tools-library.html>

Chapter 1. DB2 Cloning Tool overview

DB2 Cloning Tool for z/OS (also referred to as DB2 Cloning Tool) makes it easy to quickly clone a DB2 subsystem or a DB2 table space.

DB2 Cloning Tool can also be used to clone non-DB2 volumes.

What's new in DB2 Cloning Tool

This topic summarizes the technical changes for this edition.

New and changed information is indicated by a vertical bar (|) to the left of a change. Editorial changes that have no technical significance are not noted.

SC19-3493-03, February 2015

New information has been added for Version 3, Release 1. The documentation changes for this release are listed in this section.

- Instructions for configuring DB2 Cloning Tool in an ACF2 environment were added to the topic “Set up your environment prior to customization” on page 18.
- **DB2 Cloning Tool Subsystem Cloning**
 - The DB2UTILXCLEAN command was added. This optional command allows you to remove utility information from the target DB2 subsystem. The following topics were added or modified, and messages were added to support this feature:
 - “DB2UTILXCLEAN” on page 420
 - “DB2UTILXCLEAN command syntax” on page 421
 - “DB2UTILXCLEAN command and keyword definitions” on page 421
 - “DB2UTILXCLEAN step JCL example” on page 422
 - “Step JCL example to identify objects with UT status” on page 423
 - “DB2UTILXCLEAN command defaults” on page 287 for the ISPF interface
 - “Parameter files and parameter descriptions” on page 124
 - Chapter 9, “Cloning DB2 subsystems,” on page 95
 - “Cloning a DB2 subsystem” on page 80
 - “DB2 offline cloning” on page 95
 - “DB2 offline cloning procedure” on page 96
 - “DB2 offline cloning with removal of data sharing members procedure” on page 98
 - “DB2 offline cloning with target becoming non-data sharing procedure” on page 101
 - “DB2 online cloning” on page 104
 - “DB2 online cloning procedure” on page 105
 - “DB2 online cloning with removal of data sharing members procedure” on page 111
 - “DB2 online cloning with target becoming non-data sharing procedure” on page 117

- “DB2 subsystem cloning from a DB2 BACKUP SYSTEM backup” on page 538
- “DB2 subsystem cloning to a specific point in time from a DB2 BACKUP SYSTEM backup” on page 547
- “DB2 subsystem cloning from other system level backups when backup volumes are online” on page 566
- “DB2 subsystem cloning from other system level backups when backup volumes are offline” on page 570
- A note stating that certain commands must use the same value for DB2-NAME was modified throughout the user guide to include DB2UTILXCLEAN as one of those commands.
- The RESTORE-FROM-DUMPTAPES optional command enables subsystem cloning directly from DB2 system-level backup dump tapes. The stored procedure was also modified to clone from dump tapes. The following topics were added or modified, and messages were added to support this feature:
 - “Function authorization requirements” on page 20
 - Chapter 23, “DB2 Cloning Tool Subsystem Cloning commands,” on page 337
 - “RESTORE-FROM-DUMPTAPES” on page 450
 - “RESTORE-FROM-DUMPTAPES command syntax” on page 450
 - “RESTORE-FROM-DUMPTAPES command and keyword definitions” on page 451
 - “RESTORE-FROM-DUMPTAPES step JCL example” on page 454
 - “DB2GETBACKINFO” on page 385
 - “DB2GETBACKINFO command syntax” on page 385
 - “DB2GETBACKINFO command and keyword definitions” on page 386
 - “DB2GETBACKINFO step JCL examples” on page 388
 - “Parameter files and parameter descriptions” on page 124
 - “Stored procedure example: Cloning from a system-level backup” on page 140
 - “Stored procedure example: Cloning from DB2 system-level backup dump tapes” on page 141
 - “Stored procedure example: Cloning from DB2 system-level backup dump tapes across sysplexes” on page 143
 - “Setting subsystem cloning defaults” on page 279 for the ISPF interface
 - “RESTORE-FROM-DUMPTAPES command defaults” on page 288 for the ISPF interface
 - “Step summary for subsystem cloning from a system-level backup” on page 318 for the ISPF interface
 - “DB2 subsystem cloning from DB2 BACKUP SYSTEM dump tapes” on page 559
- The topic “DB2SQL” on page 399 was updated to describe potential issues with user-defined indexes on the DB2 catalog, and messages were added to help identify these issues.
- Keywords were added to the BCSCLEAN command to allow you to remove orphaned target catalog entries and other data sets from the target catalog. The CLEANUP-CATALOG-ORPHANS and CLEANUP-CATALOG-DSNMASKS keywords were added to following topics, and messages were added to support this feature:

- "BCSCLEAN" on page 345
- "BCSCLEAN command syntax" on page 346
- "BCSCLEAN command and keyword definitions" on page 346
- "BCSCLEAN step JCL examples" on page 349
- "Function authorization requirements" on page 20
- "Parameter files and parameter descriptions" on page 124
- **DB2 Cloning Tool Table Space Cloning**
 - DB2 Cloning Tool Table Space Cloning can now use image copies as the source for table space and index cloning. The following topics were added or modified, and messages were added to support this feature:
 - Chapter 18, "Using image copies to clone table spaces and index spaces," on page 259
 - "SET command defaults" on page 293
 - "Specifying LOG-APPLY defaults" on page 311
 - "COPY command syntax" on page 474
 - "COPY command and keyword definitions" on page 476
 - "SET command syntax" on page 516
 - "SET command and keyword definitions" on page 517
 - The LOG-APPLY process can now be used when performing table space cloning between DB2 subsystems on different LPARs or on different z/OS sysplexes. Topics were updated to describe the changes to the source and target jobs, updates to the process for rebuilding indexes, and the new source TCP/IP server job, which is required for cross-LPAR log apply. The following topics were added or modified, and messages were added to support this feature:
 - Chapter 19, "Using LOG-APPLY to make consistent copies of table spaces and index spaces," on page 265
 - "SET command defaults" on page 293
 - "Specifying LOG-APPLY defaults" on page 311
 - "COPY command syntax" on page 474
 - "COPY command and keyword definitions" on page 476
 - "SET command syntax" on page 516
 - "SET command and keyword definitions" on page 517
 - The UTILITY-COMMAND-EXECUTE-PERCENT SET command was added to specify the percentage of objects or data sets that are eligible for a particular DB2 utility to be run in a single call to that utility. The topics "SET command syntax" on page 516 and "SET command and keyword definitions" on page 517 were updated and a message was added for this feature.
 - The OBJECT-MISMATCH-RETURN-CODE COPY command was added to allow you to change the return code associated with a particular object mismatch. The topics "COPY command syntax" on page 474 and "COPY command and keyword definitions" on page 476 were updated. The ISPF interface topic "Specifying OBJECT-MISMATCH-RETURN-CODE defaults" on page 315 was added. Messages were also added for this feature.
 - The NULLSTORCLASS keyword was added to the DATA-MOVER parameter of the COPY command. NULLSTORCLASS allows the target data set SMS storage class to be assigned using the ACS routines rather than using the source data set SMS storage class. The topics "COPY command syntax" on page 474

page 474 and “COPY command and keyword definitions” on page 476 were updated. The ISPF interface topic “COPY command defaults” on page 299 also was updated.

- The DSS-COPY-COMMANDS keyword was clarified for copying a large number of data sets; refer to “COPY command and keyword definitions” on page 476 and “CKZINI keyword syntax and descriptions” on page 577.
- The SET command PARTLEVEL parameter was updated to indicate that PARTLEVEL should not be specified when generating DDL; see “LISTDEF command and keyword definitions” on page 514.
- The NULLSTORCLASS keyword was added to the DATA-MOVER parameter of the COPY command. NULLSTORCLASS allows the target data set SMS storage class to be assigned using the ACS routines rather than using the source data set SMS storage class. The topics “COPY command syntax” on page 474 and “COPY command and keyword definitions” on page 476 were updated. The ISPF interface topic “COPY command defaults” on page 299 was updated.
- The REBUILD-INDEXES-EXECUTE keyword in “SET command and keyword definitions” on page 517 and “SET command defaults” on page 293 in the ISPF interface chapter were updated to clarify DD requirements when using this keyword.
- **Changes to the ISPF interface**
 - The “Table space cloning DD descriptions” on page 291 and the “Setting default DD specifications for table space cloning” on page 289 topics were updated with new DDs SYSINCKZ and SYSOUT.
 - The attribute table in “Specifying DDL attribute change defaults” on page 309 was updated.
 - A new panel for specifying source and target subsystems was added to the interface. The topic “Select the source and target DB2 subsystems” on page 328 was added and the “Table space cloning steps summary” on page 328 was updated.
 - The job reference in “Subsystem cloning job reference” on page 323 was updated.
 - Table space cloning steps in the “Table space cloning steps summary” on page 328 and the “Build the table space cloning jobs from a profile” on page 331 topics were updated.

SC19-3493-02, October 2013

New information has been added for Version 3, Release 1. The documentation changes for this release are listed in this section.

- Updates were made to the customization process for DB2 V11 support. A new log apply table upgrade task was added to the Tools Customizer Product Parameters panel; the topic “Worksheets: Gathering parameter values for Tools Customizer” on page 28 was updated with parameter information and a description of the generated job.
- The topics in Chapter 2, “Preparing to customize DB2 Cloning Tool,” on page 17 were expanded to include an end-to-end customization checklist and planning worksheets to facilitate customization of DB2 Cloning Tool. In addition, the topics formerly located in the Customization Reference were moved into this chapter.

DB2 Cloning Tool Subsystem Cloning

- Updates for DB2 Version 11 support are as follows:

- Support was added for the new RESTORE SYSTEM LOGONLY SWITCH VCAT keyword. The support includes the ability to clone only the log volumes of a DB2 system level backup (SLB), and to generate the input needed by the RESTORE SYSTEM SYSVALUEDDN file. The following enhancements were implemented:
 - The USERSGDEFS-DDN keyword was added to BACKINFO-REFORMAT command. The topics “BACKINFO-REFORMAT” on page 338, “BACKINFO-REFORMAT command syntax” on page 339, and “BACKINFO-REFORMAT command and keyword definitions” on page 340 were updated.
 - The LOGONLY and DATABASESONLY keywords were added to DB2GETBACKINFO command. The topics “DB2GETBACKINFO command syntax” on page 385 and “DB2GETBACKINFO command and keyword definitions” on page 386 were updated.
 - The SYSVALUE-DDN keyword was added to the DB2UPDATE command. The topics “DB2UPDATE command syntax” on page 412 and “DB2UPDATE command and keyword definitions” on page 413 were updated.
 - The SOURCE-STORAGEGROUP parameter was added to the stored procedure cloning parameter file. The topic “Parameter files and parameter descriptions” on page 124 was updated.
 - The support for RESTORE SYSTEM LOGONLY SWITCH VCAT makes it possible to clone to a specific point in time that is different from the point in time when the BACKUP SYSTEM was taken. A new cloning scenario was added to explain this procedure; see the topic “DB2 subsystem cloning to a specific point in time from a DB2 BACKUP SYSTEM backup” on page 547.
- The topics “DB2FIX command and keyword definitions” on page 381, “DB2SETLOG command and keyword definitions” on page 397, “DB2SQL command and keyword definitions” on page 400, “DB2START command and keyword definitions” on page 405, and “DB2STOP command and keyword definitions” on page 409 were updated to clarify that a group name should not be used when specifying a DB2 subsystem.

DB2 Cloning Tool Table Space Cloning

- Changes for DB2 Version 11 include:
 - The REPAIR utility in DB2 V11 has been enhanced to detect and correct catalog inconsistencies. Inconsistencies might occur when DB2 objects are copied from one subsystem to another. New keywords were added to the COPY and SET commands to generate REPAIR utility jobs with REPAIR CATALOG and REPAIR VERSION commands. The topics “SET command syntax” on page 516, “SET command and keyword definitions” on page 517, “COPY command syntax” on page 474, “COPY command and keyword definitions” on page 476, and “Job templates for utilities” on page 272 were updated. These commands were also added to the ISPF interface; the topics “COPY command defaults” on page 299 and “SET command defaults” on page 293 were updated.
 - New utility job templates for REPAIR and REORG TABLESPACE (when converting from 6-byte to 10-byte RBA/LRSN) were added to the topic “Job templates for utilities” on page 272.
- Support for the following commands was added to the ISPF interface: ALWAYS-COPY-HISTORY-TABLES, COPY-IJ-TO-NONEXISTENT-TARGET, EXTEND-TARGET-PBG-TABLESPACE, DB2 10 OR LATER, WARN-ON-

DATASET-EXTENSION-MISMATCH, and SUBTASK-DATASET-EXTENSIONS. The following topics were updated: "COPY command defaults" on page 299 and "SET command defaults" on page 293.

- The text of the COPY command COPY-IF-NO-DB2-TARGET-OBJECTS keyword was updated to clarify how target objects created with DEFINE NO are handled. Refer to the topic "COPY command and keyword definitions" on page 476.
- The text of the COPY command CHECK-DATASET-COMPATIBILITY keyword was updated to clarify how target objects created with DEFINE NO are handled. Refer to the topic "COPY command and keyword definitions" on page 476.
- The &&SRCOBJS and &&TRGOBJS template variables were added to the topic "Predefined processing variables" on page 269.

SC19-3493-01, May 2013

New information has been added for Version 3, Release 1. The documentation changes for this release are listed in this section.

- You can add DB2 Cloning Tool to the DB2 Administration Tool Launchpad; refer to "Adding DB2 Cloning Tool to the DB2 Administration Tool Launchpad" on page 72.
- Several DB2 Cloning Tool JCL samples were updated to correct PARMLIB and load library data set names.
- The ISPF interface requires a region of 30000 KB; this information was added to Prerequisites and "The DB2 Cloning Tool ISPF interface" on page 275.
- The topic "Product parameters, steps, and tasks for DB2 Cloning Tool was updated to explain that when customizing DB2 Cloning Tool using Tools Customizer, the CLIST library that will contain the ISPF interface CLIST must already exist.

DB2 Cloning Tool Subsystem Cloning

- In the topic Prerequisites, updated the z/OS restriction to z/OS 1.12 and later.
- New keywords were added to allow DB2START to end with a return code if DB2 terminates before becoming active, instead of waiting for the time limit to expire. The MSTR-DETECT-WAIT and STOP-WAITING-IF-DB2-STOP keywords have been added to "DB2START command and keyword definitions" on page 405. Several messages were added or changed for this addition.
- A new keyword was added to allow DB2START to stop waiting and provide a return code if a DSNR020I WTOR is issued. The STOP-WAITING-IF-DSNR020I keyword was added to the topic "DB2START" on page 404. Updates were also made to the topics "DB2ALTERBSDS" on page 374 and "DB2 subsystem cloning from a DB2 BACKUP SYSTEM backup" on page 538. Several messages were added or changed for this addition.
- For emphasis, a statement about ensuring that no utilities are registered in SYSUTILX when the source DB2 subsystem is cloned was made into an important note, in multiple locations in "DB2 offline cloning" on page 95 and "DB2 online cloning" on page 104.
- The step JCL examples in the topic Chapter 23, "DB2 Cloning Tool Subsystem Cloning commands," on page 337 were updated to include the location of sample JCL for the commands.
- Parameters listed as input to the ACS command in the DRIVEACS keyword for the RENAME command were incorrect and were corrected; the topic "RENAME command and keyword definitions" on page 436 was updated.

DB2 Cloning Tool Table Space Cloning

- New utility job templates allow you generate DB2 utility jobs that can be run on the cloned target objects. Refer to “Job templates for utilities” on page 272. In addition, the topic “Specifying default job template variables” on page 298 was updated to specify reserved variables.
- You can now invoke DB2 Cloning Tool directly from the IBM® DB2 Administration Tool for z/OS interface. Use DB2 Admin to select the table spaces to clone, then use DB2 Cloning Tool to clone the spaces. Refer to the topic Chapter 22, “Cloning table spaces from DB2 Administration Tool,” on page 335.
- The TARGET-JOB-INDEX-REBUILD-DDN parameter was added to “COPY” on page 473. This parameter can be used to rebuild all indexes whose tables were affected by data masking or log apply page changes. In addition to the new keyword, the topics Chapter 17, “Using data masking with table space cloning,” on page 239 and Chapter 19, “Using LOG-APPLY to make consistent copies of table spaces and index spaces,” on page 265 were updated with new instructions for rebuilding indexes. The parameter was also added to the ISPF interface topic “COPY command defaults” on page 299 and new messages were added for this feature.
- CHECK-DATASET-COMPATIBILITY was added to check several VSAM data set attributes between source and target for compatibility. This parameter was added to “COPY” on page 473. The parameter was also added to the ISPF interface topic “COPY command defaults” on page 299 and messages were added to support this feature.
- IGNORE-RF-MISMATCH-IF-NO-VAR-COLS was added to allow copy of table spaces from source to target when there is a mismatch involving reordered row format. This parameter was added to “COPY” on page 473. The parameter was also added to the ISPF interface topic “COPY command defaults” on page 299 and a message was added to support this feature.
- EXTEND-TARGET-PBG-TABLESPACE was added to add partitions to target partition-by-growth table spaces when the source table space has more partitions than the target. This parameter was added to “COPY” on page 473 and the topic “Considerations for DB2 Version 9.1 and later” on page 158 was updated.
- COPY-IJ-TO-NONEXISTENT-TARGET was added to create data sets on the target subsystem without the need to rename and delete temporary data sets on the target. This parameter was added to “COPY” on page 473. Several messages were added for this feature.
- WARN-ON-DATASET-EXTENSION-MISMATCH was added to specify a warning or informational message if a mismatch in the number of extensions between source and target data sets is encountered. This parameter was added to “COPY” on page 473. Messages were added and changed for this feature.
- ALWAYS-COPY-HISTORY-TABLES was added to select history tables associated with all objects specified on a LISTDEF. This parameter was added to “COPY” on page 473. Messages were added and changed for this feature.
- HISTORY was added to allow only history tables to be processed on a LISTDEF statement. This parameter was added to “LISTDEF” on page 512.
- CONNECT-DB2-ON-CLIENT-CONNECT was added to allow the TCP/IP server job to connect to DB2 only when a client connect occurs. This parameter was added to “SET” on page 516.
- SUBTASK-DATASET-EXTENSIONS was added to allow data set extents to be processed in any available subtask. This parameter was added to “SET” on page 516. A message was added for this feature.
- When using LOG-APPLY, the source and target DB2 must be on the same LPAR; this restriction was added to Chapter 19, “Using LOG-APPLY to make consistent

copies of table spaces and index spaces,” on page 265 and the LOG-APPLY keyword in “COPY command and keyword definitions” on page 476.

- Added note about using SET REMOTE-CONNECT-TYPE to “DDF and TCP/IP considerations” on page 153.
- DDL processing cannot be simulated; this information was noted in “Considerations for generating target object DDL using PROCESS-DDL” on page 160, “COPY command defaults” on page 299, and “COPY command and keyword definitions” on page 476.
- A sentence was added to clarify that certain objects must be created by the user as part of table space cloning. The following topics were updated: “Step 1: Create target DB2 table spaces and index spaces in the target DB2 catalog” on page 179 and “Step 1: Verify/create target DB2 table spaces and index spaces in target DB2 catalog” on page 199.
- Clarifications about IP and port parameters used with the table space cloning jobs were added to the ISPF interface topic “COPY command defaults” on page 299 and the command reference topic “COPY command and keyword definitions” on page 476.
- The LISTDEF keyword CLONED was incorrectly specified as CLONE in previous editions and has been corrected; refer to the topic “LISTDEF” on page 512.
- Incorrect applitotypes for CLOSE were modified in the topic “DDL-ATTRIBUTE-CHANGE parameter values” on page 506.
- The maximum value for the TCPIP-SERVER-PORT was corrected in “SET command defaults” on page 293, “SET command and keyword definitions” on page 517 and message CKZ53731E.
- Corrected the default value for INCLUDE-ALL-RI in “COPY” on page 473.

SC19-3493-00, October 2011

New information has been added for Version 3, Release 1. The documentation changes for this release are listed in this section.

- DB2 Cloning Tool V3.1 is customized using IBM Tools Customizer for z/OS, a component that simplifies and consolidates many of the processes required to customize DB2 Cloning Tool. If you are responsible for installing and customizing DB2 Cloning Tool, before proceeding with the customization you should review [Starting and preparing Tools Customizer for use](#). The topic [Chapter 24, Customization Reference](#) provides information about the customization tasks and parameters for DB2 Cloning Tool. In addition, the following topics were added:
 - [Tools Customizer overview](#)
 - [Tools Customizer terminology](#)
 - [Data sets that Tools Customizer uses during customization](#)
 - [Tools Customizer messages](#)
 - [Tools Customizer troubleshooting](#)
 - Chapter 5, “After customizing DB2 Cloning Tool using Tools Customizer,” on page 71
- A stored procedure has been provided that can be called to perform a DB2 subsystem cloning. The stored procedure generates the necessary jobs to do the subsystem cloning, schedules the jobs in the DB2 administrative task scheduler, and monitors the execution of the jobs. Refer to the topic [Chapter 10, “Subsystem cloning using the DB2 Cloning Tool stored procedure,”](#) on page 123.

- When work file databases are cloned for a data sharing environment, they are now usable by the target members. Manually dropping and recreating the work file databases is no longer required as part of the cloning process. The procedures were updated in Chapter 9, “Cloning DB2 subsystems,” on page 95 to reflect this change.
- Support for Preserve Mirror FlashCopy[®] was added to DB2 Cloning Tool Subsystem Cloning. The COPY command was updated to allow specifying the desired Preserve Mirror FlashCopy options. Refer to the topic “COPY” on page 350.
- Support for subsystem cloning using a system level backup was added to the ISPF interface. For information, refer to the topic “Specify source and target volume pairings” on page 321.
- User DDs for table space cloning are now entered on a separate panel in the ISPF interface. Refer to “Setting default DD specifications for table space cloning” on page 289.
- The LOG-APPLY keyword was added to the DB2 Cloning Tool Table Space Cloning COPY command. This keyword allows the DB2 log records written between the time that source job copies are done and the target job is run to be applied to the target objects. This feature eliminates the need to stop and start source objects to achieve a consistent copy. The topic Chapter 19, “Using LOG-APPLY to make consistent copies of table spaces and index spaces,” on page 265 was added and the keyword was added to the topic “COPY” on page 473.
- DB2 Cloning Tool Table Space Cloning can optionally generate and execute DDL to be used for creating missing target objects. Missing objects are defined by using the source DDL to create the target DDL, and the PROCESS-DDL keyword allows you to easily change attributes of the target objects during the copy. The PROCESS-DDL keyword was added to the COPY command and topics were added to describe how to use this feature. For information, Refer to “Considerations for generating target object DDL using PROCESS-DDL” on page 160, “COPY” on page 473, and “DDL-ATTRIBUTE-CHANGE parameter values” on page 506. The ISPF interface was also updated; see Chapter 21, “Using the ISPF interface,” on page 275.
- Support for Preserve Mirror FlashCopy was added to DB2 Cloning Tool Table Space Cloning. The COPY command was updated to allow specifying the desired Preserve Mirror FlashCopy options. Refer to the topic “COPY command and keyword definitions” on page 476.
- The STOGROUP parameter was added to the DB2 Cloning Tool Table Space Cloning LISTDEF command to allow all the objects in all the databases in a DB2 storage group to be selected with a single LISTDEF statement. Refer to the topic “LISTDEF” on page 512.
- The LONGVAR-COMPATIBILITY keyword was added to the DB2 Cloning Tool Table Space Cloning COPY command. This keyword prevents a mismatch message when running a source job in which the source objects have LONGVAR columns and the target objects have corresponding VARCHAR columns (or vice versa). The keyword was added to the topic “COPY” on page 473. The ISPF interface was also updated; see Chapter 21, “Using the ISPF interface,” on page 275.
- Two keywords were added to the DB2 Cloning Tool Table Space Cloning SET command. The USE-RUNTIME REPOSITORY keyword allows a failed target job to be re-run to only process data sets that have not been processed. The REPORT-JOB keyword allows a report to be generated based on data in the

runtime repository. These keywords were added to "SET" on page 516. The ISPF interface was also updated; see Chapter 21, "Using the ISPF interface," on page 275.

- When using data masking for DB2 Cloning Tool Table Space Cloning, masking of DECFLOAT column data is now supported for STATIC, SCRAMBLE and USEREXIT rules. The topic Chapter 17, "Using data masking with table space cloning," on page 239 was updated.
- You can route warning and error messages for DB2 Cloning Tool Table Space Cloning to a separate DD. Refer to "Message output" on page 158.
- A new DATASUBTYPE command was added to DB2 Cloning Tool Table Space Cloning that allows you to specify the data subtype value (BIT, MIXED, or SBCS) needed by data masking. Refer to the topic "DATASUBTYPE" on page 510 for information. Messages were added to support this feature.
- A new DB2 Cloning Tool Table Space Cloning parameter makes additional checks on index compatibility when cloning table spaces. The CHECK-INDEX-KEYS parameter was added to "COPY" on page 473. Messages were added to support this feature. The ISPF interface was also updated; see Chapter 21, "Using the ISPF interface," on page 275.
- WARN-ON-SIMPLE-TABLESPACE was added to DB2 Cloning Tool Table Space Cloning to allow you to be warned when a simple table space is to be copied. This parameter was added to "COPY" on page 473. Messages were added to support this feature. The ISPF interface was also updated; see Chapter 21, "Using the ISPF interface," on page 275.
- The topic "Considerations for DB2 Version 9.1 and later" on page 158 was revised to:
 - Clarify how DB2 Cloning Tool Table Space Cloning handles partition-by-growth table spaces.
 - Add a section describing considerations for reordered row format in DB2 Version 9.1 NFM and later.
- The topics "SET command and keyword definitions" on page 517 and "Specifying ADVISORY-STATUS-VALUES defaults" on page 296 were updated to add AREOR as a possible ADVISORY-STATUS-VALUE for DB2 10 and later.

What does DB2 Cloning Tool do?

DB2 Cloning Tool has two components: DB2 subsystem cloning and DB2 table space cloning.

DB2 subsystem cloning:

- Clones automatically by using IBM FlashCopy, STK SnapShot, or EMC TimeFinder/Clone Mainframe Snap Facility's volume level support
- Uses any volume fast replication or onsite mirror tool

The cloned DB2 subsystem can be accessed from the same z/OS system. This increases productivity in several ways:

- Significantly reduces production online downtime and the costs associated with cloning a DB2 subsystem with traditional tools.
- Provides more availability and services to customers because DB2 no longer needs to be shut down or conditioned the long traditional way.
- Uses less personnel time to clone a DB2 subsystem – what used to take hours or days now takes just minutes.
- Allows management of larger storage environments with the same staff.

- Provides quicker throughput and faster turnaround time.
- Provides virtually 24x7 access to data.
- Creates fast quality assurance and/or test environments.

Volume fast replication tools allow a group of data to be replicated within minutes to provide duplicate environments. However, these clones have an inherent problem: the internal volume name, volume internals, and all data set names reflect the source volume name.

DB2 Cloning Tool quickly solves this problem. Once the data is cloned or replicated, DB2 Cloning Tool renames and catalogs the data sets on the cloned volumes, fixes the volume internals, and updates the DB2 internals so the cloned DB2 subsystem can be accessed from the same MVS™ system. DB2 Cloning Tool provides a vast improvement over existing methods. It supports entire DB2, DB2 PeopleSoft, and DB2 SAP subsystems in either online or offline mode.

DB2 table space cloning:

- Clones automatically by using IBM FlashCopy, STK SnapShot, or EMC TimeFinder/Clone Mainframe Snap Facility's data set level support
- Uses any data set copy, fast or slow

The cloned DB2 table spaces can be cloned within the same or different z/OS system. This increases productivity in several ways:

- Uses less personnel time to copy DB2 table spaces and associated manual tasks – what used to take hours now takes just minutes.
- Provides quicker throughput and faster turnaround time.
- Provides virtually 24x7 access to data.
- Creates fast refreshes of quality assurance and/or test environments

The remainder of this introductory topic addresses only DB2 subsystem cloning using volume copies. For more information about DB2 Cloning Tool table space cloning, refer to Chapter 11, “DB2 Cloning Tool Table Space Cloning overview,” on page 147.

Cloning definitions

A clone is an exact copy, indistinguishable from the original.

Cloning is the act of replicating data, making it accessible, and then using the replica in lieu of the original data for other purposes. Replication tools clone the data by volume, and DB2 Cloning Tool makes the clone accessible.

Why clone a DB2 subsystem?

There are several reasons to clone a DB2 subsystem.

Clone a DB2 subsystem:

- To create a production quality assurance environment
- To move a group of end users to the cloned DB2 subsystem to lessen the performance impact on the production system
- To give end users access to an application that is updated on a continual basis
- To allow developers to begin changing the application for the next phase of code updates
- To test new functions and features of SAP, or PeopleSoft

- To run an online inquiry while batch runs
- For data mining
- For data warehousing

Can I clone a DB2 subsystem without DB2 Cloning Tool?

A DB2 subsystem can be cloned without using DB2 Cloning Tool but it is a complicated process that can take days, and requires using target volumes on a separate LPAR.

By using DB2 Cloning Tool, you can clone a DB2 subsystem within minutes instead of hours or days. Also, DB2 Cloning Tool does not require a separate LPAR.

DB2 support

A DB2 subsystem can be cloned in either an offline environment (DB2 is stopped and started) or an online environment (using DB2 SUSPEND and RESUME, or via consistent FlashCopy, SnapShot, or TimeFinder/Clone, or via consistent split or break mirror). There is no requirement for a separate LPAR.

DB2 Cloning Tool updates DB2 internal control information in the BSDS, directory, and DB2 catalog to reflect the target DB2 subsystem names.

DB2 data-sharing groups and non data-sharing DB2 subsystems are supported. When cloning a DB2 data-sharing group, you can optionally reduce the number of DB2 data-sharing members, or go from data-sharing to non data-sharing.

Volume copy products supported by DB2 Cloning Tool

DB2 Cloning Tool will rename and catalog data sets on target volumes created with any type of replication mechanism, where target volumes are exact replicas of source volumes.

If target volumes still have the source volume serial number (VOLSER), and are varied offline, DB2 Cloning Tool can re-label and vary the target volumes online.

Restriction: "Exact Replicas": The only exception to a target volume being an exact replica of its source volume counterpart is the internal volume serial number. Copies already partially modified are not acceptable – such as volume snaps via SIBBATCH where the SYS1.VVDS and SYS1.VTOCIX names may have already been modified.

Storage blades

The fast replication copy services outlined in this section can be used to create the volume copies used for DB2 system clones, or to create the data set copies used for table space and index space refreshes. A storage blade represents fast replication copy services invoked directly by DB2 Cloning Tool.

DB2 Cloning Tool executes the DFSMSdss blade to issue IBM FlashCopy or STK SnapShot copy commands either by volume or by data set. When cloning DB2 systems or table spaces that reside on EMC DASD, DB2 Cloning Tool uses the EMC API to invoke TimeFinder/Clone to copy the data by volume or by data set. When cloning DB2 subsystems that reside on EMC DASD and use TimeFinder/Mirror, or that reside on Hitachi Storage Systems, an appropriate

process is performed before DB2 Cloning Tool cloning automation is invoked and a list of copied storage volumes are passed to DB2 Cloning Tool for use in later processing steps.

IBM storage blades

- IBM DFSMSdss blade
- ADRDSSU utility invoked
- Fast replication (preferred)
- By volume or data set
- FlashCopy V2 (IBM, EMC, HDS)
- SnapShot (STK, Ramac)

EMC storage blades

- EMC TimeFinder
- TimeFinder/Clone Mainframe Snap Facility's volume level support
- EMC Consistency Technology
- TimeFinder/Clone Mainframe Snap Facility's data set level support

IBM FlashCopy, STK SnapShot, and EMC TimeFinder/Clone allow the creation of what appears to be a copy of a volume, in a short period of time. The target volume of a copy will be a mirror image of the source volume, with the exception of the internal 'VOLSER', which remains as is, if the COPYVOLID option is not specified. DB2 Cloning Tool requires the target VOLSER to remain as is, which allows the target volume to remain varied online after the copy initiation, and therefore negates the need for a CLIP (re-label) and VARY ON.

Because the logical image of a target volume may be modified before the copy is actually complete, the DB2 Cloning Tool rename step may be executed while the background copy is still in progress. Likewise, should the DB2 Cloning Tool process complete before a volume is completely copied, the application may also begin using volumes before the background copy is complete.

Onsite mirror tools by volume

Other cloning mechanisms, such as IBM PPRC, Hitachi ShadowImage, EMC TimeFinder/Mirror, Softek Replicator, or Innovation Data Processing FDRPAS, allow the creation of a point-in-time image of a volume by establishing a mirror, and then splitting the mirror once the target volume is synchronized with the source volume. At the time of this publication, establishing and splitting mirrors must be accomplished with user-provided steps prior to execution of DB2 Cloning Tool.

DB2 Cloning Tool features and benefits

DB2 Cloning Tool provides solutions to many different types of problems.

DB2 Cloning Tool offers several unique and significant features that you can use to quickly clone your DB2 data for use in testing or other requirements:

- Quickly fixes volume conflicts (VTOC, VTOCIX and VVDS) and then renames and re-catalogs the target data sets to solve the data access problem
- Offers extended rename capability to support DB2 log and BSDS desired names
- DB2 online or offline cloning in minutes instead of days, without a separate image

- Automatic updating of DB2 internals to reflect renamed data sets
- DB2 data sharing
- DB2 data sharing many to less members
- DB2 support for either DB2 data sharing or from data sharing to non-data sharing for maximum flexibility
- Provides automatic pairing of volume characteristics (SMS and non-SMS, or by device size) that DFSMSdss doesn't do
- Allows FlashCopy, SnapShot, or TimeFinder/Clone by VOLSER masks or entire storage groups or any combination to eliminate the requirement for individual volume specification
- Provides extensive SMS options that enable you to determine how the SMS class constructs will be applied to your cloned data sets to ensure they are managed correctly
- Issuance of DFSMSdss commands or EMC TimeFinder/Clone commands within the product for ease of use
- Source volume ICF catalog information collection concurrent with FlashCopy, SnapShot, or TimeFinder/Clone initiation for rename integrity
- Early resumption of source volume activity to reduce outages
- Enhanced data set rename masking characters for flexibility
- User options to decide disposition of 'abnormal' data sets and catalog entries
- Multivolume data set and VSAM sphere integrity checks to prevent orphaned data
- Rename into existing populated ICF catalog(s) or a specific one
- Faster cataloging than conventional means
- Test for termination of FlashCopy, SnapShot, or TimeFinder/Clone relationships
- Mechanism to remove orphaned catalog entries from previous executions
- Simulate modes for most commands

Service updates and support information

Service updates and support information for this product, including software fix packs, PTFs, frequently asked questions (FAQs), technical notes, troubleshooting information, and downloads, are available from the web.

To find service updates and support information, see the following website:

http://www.ibm.com/support/entry/portal/Overview/Software/Information_Management/DB2_Tools_for_z~OS

Product documentation and updates

DB2 Tools information is available at multiple places on the web. You can receive updates to DB2 Tools information automatically by registering with the IBM My Notifications service.

Information on the web

The DB2 Tools Product Documentation web page provides current product documentation that you can view, print, and download. To locate publications with the most up-to-date information, refer to the following web page:

<http://www.ibm.com/software/data/db2imstools/db2tools-library.html>

You can also access documentation for many DB2 Tools from IBM Knowledge Center:

<http://www.ibm.com/support/knowledgecenter>

Search for a specific DB2 Tool product or browse the **Information Management > DB2 for z/OS family**.

IBM Redbooks® publications that cover DB2 Tools are available from the following web page:

<http://www.redbooks.ibm.com>

The Data Management Tools Solutions website shows how IBM solutions can help IT organizations maximize their investment in DB2 databases while staying ahead of today's top data management challenges:

<http://www.ibm.com/software/data/db2imstools/solutions/index.html>

Receiving documentation updates automatically

To automatically receive emails that notify you when new technote documents are released, when existing product documentation is updated, and when new product documentation is available, you can register with the IBM My Notifications service. You can customize the service so that you receive information about only those IBM products that you specify.

To register with the My Notifications service:

1. Go to <http://www.ibm.com/support/mysupport>
2. Enter your IBM ID and password, or create one by clicking **register now**.
3. When the My Notifications page is displayed, click **Subscribe** to select those products that you want to receive information updates about. The DB2 Tools option is located under **Software > Information Management**.
4. Click **Continue** to specify the types of updates that you want to receive.
5. Click **Submit** to save your profile.

How to send your comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this book or any other IBM product documentation, use one of the following options:

- Use the online reader comment form, which is located at <http://www.ibm.com/software/data/rcf/>.
- Send your comments by email to comments@us.ibm.com. Include the name of the book, the part number of the book, the version of the product that you are using, and, if applicable, the specific location of the text you are commenting on, for example, a page number or table number.

Accessibility features

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use a software product successfully.

The major accessibility features in this product enable users to perform the following activities:

- Use assistive technologies such as screen readers and screen magnifier software. Consult the assistive technology documentation for specific information when using it to access z/OS interfaces.
- Customize display attributes such as color, contrast, and font size.
- Operate specific or equivalent features by using only the keyboard. Refer to the following publications for information about accessing ISPF interfaces:
 - *z/OS ISPF User's Guide, Volume 1*
 - *z/OS TSO/E Primer*
 - *z/OS TSO/E User's Guide*

These guides describe how to use the ISPF interface, including the use of keyboard shortcuts or function keys (PF keys), include the default settings for the PF keys, and explain how to modify their functions.

Chapter 2. Preparing to customize DB2 Cloning Tool

Before you start to customize DB2 Cloning Tool for the first time, determine all of the customization values that you need to specify during the customization process, and familiarize yourself with all of the customization tasks.

Checklist for customization tasks

The following checklist lists and describes each significant customization step. Use this checklist to guide you through the entire customization process.

Tip: Print the following checklist and the data set names and parameter values worksheets. Use the worksheets to record your values, and refer to them during the customization process.

Task	Link to detailed instructions	Status
Tools Customizer basics		
Before you begin the customization process, familiarize yourself with Tools Customizer terminology, data sets, and other basic information about Tools Customizer.	"Tools Customizer terminology" on page 529 and "Data sets that Tools Customizer uses during customization" on page 531 and	
Software requirements		
Verify that your environment meets the minimum software requirements. To install and use DB2 Cloning Tool, your environment must be running a supported version of the z/OS operating system and of DB2 for z/OS. Additionally, certain levels of maintenance must be applied.	"Verify that your environment meets software requirements" on page 19	
SMP/E installation		
Verify that DB2 Cloning Tool was installed correctly. DB2 Cloning Tool is installed by using standard SMP/E processing.	"Verify that DB2 Cloning Tool has been installed successfully" on page 19	
Verify that Tools Customizer for z/OS was installed correctly. Tools Customizer for z/OS is installed by using standard SMP/E processing.	"Verify that Tools Customizer for z/OS has been installed successfully" on page 19	
Security requirements		
Confirm that you have the required authorizations to use DB2 Cloning Tool.	"Verify that your environment meets security requirements" on page 19	
Function authorization requirements		
Review the level of authority required to perform certain DB2 Cloning Tool functions.	"Function authorization requirements" on page 20	
Use of the U.S.A. EBCDIC code set		
DB2 Cloning Tool uses the U.S.A. EBCDIC code set for specification and display of EBCDIC characters and for the extended ACS masking characters used for filtering. If the code tables used by your installation are different, then you may need to specify characters particular to your code table.	"Use of the U.S.A. EBCDIC code set" on page 22	
Filtering pattern masks		

Task	Link to detailed instructions	Status
Many DB2 Cloning Tool commands allow specification of powerful filtering masks. These filtering pattern masks can be specified with extended ACS masking characters, as specified in the table. Your installation may need to specify different masking characters to achieve the desired result if your code tables are different from the U.S.A. EBCDIC code set.	"Filtering pattern masks" on page 23	
Gather data set names		
During the customization process, you must specify data set names for Tools Customizer, DB2 Cloning Tool, and several other items.	"Worksheets: Gathering required data set names" on page 25	
APF authorization		
The SCKZLOAD library must be APF authorized.	"APF authorizing load libraries" on page 27	
Gather parameter values		
During the customization process, you must specify parameter values for DB2 Cloning Tool, for DB2, and for your LPAR.	"Worksheets: Gathering parameter values for Tools Customizer" on page 28	
Customize DB2 Cloning Tool		
Start Tools Customizer by running a REXX EXEC from the ISPF Command Shell panel.	"Starting Tools Customizer" on page 45	
Set up Tools Customizer user settings. If you are running Tools Customizer for the first time, you must modify several user settings for your environment. Otherwise, if the user settings that you have already established are still appropriate, skip this step.	"Modifying Tools Customizer user settings" on page 46	
Complete the steps in the appropriate customization roadmap based on the type of customization that you are performing.		
Customizing DB2 Cloning Tool for the first time Follow this roadmap if you do not have a customized version of DB2 Cloning Tool, and you need to customize it for the first time.	"Roadmap: Customizing DB2 Cloning Tool for the first time" on page 51	
Customizing a different version of DB2 Cloning Tool Follow this roadmap if you previously customized a version of DB2 Cloning Tool and want to use the same parameter values to customize a different version.	"Roadmap: Customizing a new version of DB2 Cloning Tool from a previous customization" on page 52	
Recustomizing DB2 Cloning Tool Follow this roadmap if you already customized DB2 Cloning Tool but want to change one or more parameter values.	"Roadmap: Recustomizing DB2 Cloning Tool" on page 53	

Set up your environment prior to customization

Prior to beginning the customization process, ensure that your environment meets all requirements, that you have installed all prerequisite software, and that you have considered how you want to customize optional features.

Verify that your environment meets software requirements

The operation of DB2 Cloning Tool depends on the following software levels:

- z/OS 1.11 and later.

Restriction: On z/OS 1.12 and later systems, the following restrictions apply:

- A source or target ICF catalog cannot be defined with extended addressability or EATTR(OPT).
 - The VVDS on the volume where a source or target ICF catalog resides cannot be defined with EATTR(OPT).
 - The VVDS on a source or target volume cannot be defined with EATTR(OPT).
- DB2 10 or DB2 11.
 - Any available new PTFs.
 - One or more of the cloning tools listed in section “Volume copy products supported by DB2 Cloning Tool” on page 12, or slow copy utilities, such as DFSMSdss copy or Innovation Data Processing products.
 - If you will use DB2 Cloning Tool’s stored procedure for subsystem cloning, the DB2 administrative task scheduler must be configured and available on the DB2 systems where the stored procedure will be run.

Note: Only DB2 Version 9.1 and later administrative task schedulers are supported. For DB2 Version 9.1, APAR/PTF PM02658/UK60388 must be applied. The DB2 Version 8 administrative task scheduler is not supported.

- If you will use IBM FlashCopy for copying by data set, FlashCopy version 2 is required.
- If you will use the DB2 Cloning Tool ISPF interface, a minimum region size of 30000 KB is required.

Verify that DB2 Cloning Tool has been installed successfully

See the Program Directory for IBM DB2 Cloning Tool for z/OS, GI10-8910-00, for installation instructions.

Verify that Tools Customizer for z/OS has been installed successfully

Tools Customizer for z/OS is a component of IBM Tools Base for z/OS (5655-V93), which is available free of charge. Tools Customizer for z/OS provides a standard approach to customizing IBM DB2 for z/OS Tools.

See the Program Directory for IBM Tools Base for z/OS, GI10-8819 for installation instructions.

Verify that your environment meets security requirements

You can prevent unauthorized personnel from executing DB2 Cloning Tool commands. To use DB2 Cloning Tool Table Space Cloning, specific authorizations are also required.

In addition to user authorizations, some functions require other authorizations; refer to “Function authorization requirements” on page 20.

Authorizations required for DB2 Cloning Tool are:

- DB2 Cloning Tool requires ALTER authority to both the source and target user catalogs. By protecting ALTER authority, unauthorized personnel are prevented from executing the COPY and RENAME commands.
- DB2 Cloning Tool requires UPDATE authority to the target data sets. By protecting UPDATE authority, unauthorized personnel are prevented from executing the DB2UPDATE command.
- The COPY command requires authorization to ADRDSSU ADMIN. By protecting ADRDSSU ADMIN, unauthorized personnel are prevented from executing the COPY command. This only applies for FlashCopy where ADRDSSU is used.
- The COPYCHECK command could be used by an unauthorized person using an authorized person's journal file. To prevent this, secure the journal files.

Authorizations for DB2 Cloning Tool Table Space Cloning are:

- SELECT authority on the DB2 catalog tables for both source and target subsystems
- EXECUTE authority on the CKZPLAN on both source and target subsystems
- DISPLAYDB authority on both source and target databases
- STOPDB authority on the target database
- STARTDB authority on the target database
- ALTER authority for all target tables with identity columns
- If FUZZY-COPY(Y) is used to not stop the source, and ADRDSSU is used, then RACF® authority for "TOLERATE(ENQF)" is required
- If FUZZY-COPY(N) (the default) is used, then STOPDB and STARTDB authority for the source database is required

If your site uses ACF2 to restrict TSO command use, you may need to add the TSO commands that DB2 Cloning Tool uses to the ACF2 Command Limiting table. The TSO commands that DB2 Cloning Tool uses are CKZARGSZ, CKZ2MAIN, CKZ00500, and CKZ00600.

Function authorization requirements

The level of authority required to perform certain DB2 Cloning Tool Subsystem Cloning and DB2 Cloning Tool Table Space Cloning is detailed in this topic.

For DB2 Cloning Tool subsystem or volume cloning

- DB2 Cloning Tool COPY and RENAME commands require ALTER authority to both the source and target user catalogs. DB2 Cloning Tool does NOT update the source catalog but requires ALTER authority to access the catalog via direct access.
 - The source ICF catalog(s) requires ALTER because an IDCAMS VERIFY and an IDCAMS EXAMINE INDEXTEST NODATATEST is done prior to reading it to ensure the source catalog is not broken.
 - The target ICF catalog(s) requires ALTER because it is opened and updated using a routine that manages catalog updates.
- DB2 Cloning Tool COPY command when using DATA-MOVER PGM(ADRDSSU) may require the user ID running the command to have specific access to STGADMIN profiles in the FACILITY class.
 - The ADMINISTRATOR operand is always specified when invoking DFSMSdss. To avoid WTORs, ADRDSSU ADMINISTRATOR is used to gain permission to overlay the target volume VTOCIX and/or VVDS during the

COPY process. Because the ADMINISTRATOR operand is generated, the user ID running COPY must have READ access to FACILITY class profile STGADMIN.ADR.STGADMIN.COPY.

- If FCSETGTOK is specified, READ access is needed to FACILITY class profile STGADMIN.ADR.COPY.FCSETGT if it exists.
- If FCTOPPRCPPRIMARY is specified, READ access is needed to FACILITY class profile STGADMIN.ADR.COPY.FCTOPPRCP if it exists.
- If FlashCopy is being used, READ access is needed to FACILITY class profile STGADMIN.ADR.COPY.FLASHCPY if it exists.
- If CONSISTENT(YES) is specified, READ access is needed to FACILITY class profiles STGADMIN.ADR.COPY.FCFREEZE and STGADMIN.ADR.CGCREATE if they exist.
- If the CONCURRENT option is being used, READ access is needed to FACILITY class profile STGADMIN.ADR.COPY.CNCURRENT if it exists.
- If DB2 Cloning Tool is going to relabel the target volumes, ICKDSF REFORMAT requires RACF volume access of ALTER to RACF class DASDVOL or READ access to FACILITY class profile STGADMIN.ICK.REFORMAT. DB2 Cloning Tool will relabel the target volumes as part of COPY when VOLPAIRSDEVN or VOLPAIRSDEVN-DDN is used, or as part of VOLOPTIONS when CLIP, OFFLINECLIP, or UNCLIP are used.
- DB2 Cloning Tool RENAME command invokes ICKDSF to rebuild the VTOCIX, if VTOCIX_REBUILDER = IBM is specified in CKZINI or the volume is an Extended Address Volume. ICKDSF requires RACF volume access of ALTER to RACF class DASDVOL or READ access to FACILITY class profile STGADMIN.ICK.BUILDIX. If the default, VTOCIX_REBUILDER = MSC, is specified in CKZINI, ICKDSF is not used, unless the volume is an Extended Address Volume for which ICKDSF will always be used to rebuild the VTOCIX.
- If DB2 Cloning Tool RENAME has NOTRENAMED(DELETE) in effect, READ access to FACILITY class profile STGADMIN.IGG.DLVVRNVR.NOCAT is required.
- DB2 Cloning Tool DB2UPDATE command requires authority to the target data sets.
- DB2 Cloning Tool COPYCHECK command requires UPDATE to the journal file.
- DB2 Cloning Tool FINDUCATS command invokes DCOLLECT to identify ALIAS names of the source volume data sets in order to identify the correct source user catalogs. DCOLLECT controls access to the DCOLLECT function, by issuing a security (RACF) check for a facility class profile of STGADMIN.IDC.DCOLLECT. If this profile exists, then Read authority is necessary.
- DB2 Cloning Tool DB2FIX command requires either DB2 install SYSADM or SYSADM2 authority.
- DB2 Cloning Tool DB2SQL command requires either DB2 install SYSADM or SYSADM2 authority.
- DB2 Cloning Tool DB2SETLOG command requires authorization to connect to the DB2 subsystem and to issue DB2 SET LOG commands.
- DB2 Cloning Tool DB2START command requires authorization to issue z/OS START DB2 commands and to connect to the DB2 subsystem. When the SPECIAL or MAINT keyword is used, it requires either DB2 install SYSADM or SYSADM2.
- DB2 Cloning Tool DB2STOP command requires authorization to connect to the DB2 subsystem and issue the DB2 STOP DB2 command. When the DB2 subsystem is running in maintenance mode, it requires either DB2 install SYSADM or SYSADM2.

- DB2 Cloning Tool RENAME command requires authorization to create an EMCS console and issue z/OS MODIFY CATALOG commands. Update authorization to the VVDS data sets on the target volumes is also required.
- DB2 Cloning Tool DB2UPDATE command (if DB2XCFCLEAN(Y) is specified) requires authority to clean up XCF structures. The ability to clean up XCF structures is controlled with profiles in the FACILITY class with names of IXLSTR.structurename. If profiles exist that cover the target DB2 structures, then UPDATE access is necessary.
- DB2 Cloning Tool DB2XCFCLEAN command requires authority to clean up XCF structures. The ability to clean up XCF structures is controlled with profiles in the FACILITY class with names of IXLSTR.structurename. If profiles exist that cover the target DB2 structures, then UPDATE access is necessary.
- DB2 Cloning Tool DB2GETBACKINFO command requires authority to issue HSM LIST COPYPOOL commands. The ability to issue HSM LIST COPYPOOL commands is controlled with profiles in the FACILITY class with names of STGADMIN.ARC.LC.copypoolname. If profiles exist that cover the copypoolname being used, then READ access is necessary.
- DB2 Cloning Tool RESTORE-FROM-DUMPTAPES command requires the following:
 - READ access to HSM dump tapes (for example, TAPEVOL profile HSMHSM, or READ access by data set name).
 - RACF volume access of ALTER to RACF class DASDVOL.
 - READ access to FACILITY class profile STGADMIN.ICK.REFORMAT.
 - READ access to STGADMIN.ADR.STGADMIN.RESTORE for the target volumes.
- DB2 Cloning Tool BCSCLEAN command requires READ authority to profiles in the FACILITY class with names of STGADMIN.IGG.DIRCAT and STGADMIN.IGG.DELETE.NOSCRATCH.

For DB2 Cloning Tool Table Space Cloning

- SELECT authority on the DB2 catalog tables is required for both source and target subsystems.
- EXECUTE authority on the CKZ plan is required on both source and target subsystems.
- DISPLAYDB authority is required on both source and target databases.
- STOPDB authority is required on the target database.
- STARTDB authority is required on the target database.
- ALTER authority is required for all target tables with identity columns.
- If FUZZY-COPY(Y) is used to not stop the source, and ADRDSSU is used, then RACF authority for "TOLERATE(ENQF)" is required.
- If FUZZY-COPY(N) (the default) is used, then STOPDB and STARTDB authority for the source database is required.

Use of the U.S.A. EBCDIC code set

DB2 Cloning Tool uses the U.S.A. EBCDIC code set for specification and display of EBCDIC characters and for the extended ACS masking characters used for filtering.

If the code tables used by your installation are different, then you need to enter the EBCDIC character peculiar to your code tables that results in the binary value for the EBCDIC character specified in the product manuals.

- Data supplied as input to batch programs or input to ISPF panels:
For product code shipped in binary, when specifying input where the product takes special action based on specific characters, you must enter the EBCDIC character peculiar to your code tables that results in the binary value for the EBCDIC character specified in the product manuals, according to the U.S.A. EBCDIC code set.
For example, if an exclamation mark (!) is called for, and your code tables do not translate the ! character to a hexadecimal 5A, you must enter the character that your code table will translate to a 5A.
- Distributed ISPF panels:
Do not change distributed ISPF panels. Program code may reference ISPF panel attribute bytes. A panel change that affects an attribute byte may cause a program error.
- Product output:
Depictions of product output shown in the product manuals are based on the U.S.A. EBCDIC code set. Actual output may vary if your EBCDIC code tables are different.
- Extended ACS masking characters:
Your installation may need to specify different masking characters to achieve the desired result if your code tables are different from the U.S.A. EBCDIC code set. For more information about extended ACS masking characters, see “Filtering pattern masks.”

Filtering pattern masks

Many commands allow specification of powerful filtering masks. These filtering pattern masks can be specified with extended ACS masking characters, as specified in the table.

Note: Your installation may need to specify different masking characters to achieve the desired result if your code tables are different from the U.S.A. EBCDIC code set.

Table 1. Filtering pattern masks

Mask	Hex	Description
*	5C	<p>A single asterisk (*) represents 0–<i>n</i> characters. It can be used before and/or after an item to designate a wildcard character position. You cannot use an asterisk in the middle portion of a single item (except for data set names). Use the percent sign (%) character for that purpose.</p> <p>Example:*PROD</p> <p>In this example, an item is selected if the last four characters are PROD regardless of the starting characters. Using a single asterisk character before and after an item (i.e. *ZREM*) means that item is to be selected if any character string matches ZREM in its name.</p>

Table 1. Filtering pattern masks (continued)

Mask	Hex	Description
%	6C	<p>A percent sign (%) denotes a single-character placeholder value, which can be alpha-numeric or any special character. The percent sign can be used in any position order.</p> <p>Example:CRM%%ER6</p> <p>In this example, an item is selected if it is eight characters in length, the first three characters are CRM, and the last three characters are ER6. The two middle placeholder values (%%) can be any characters.</p>
<	4C	<p>Similar to the percent sign (%) value, the "less than" character (<) denotes a placeholder value <i>for only alpha characters</i>. The < character can be used in any position order.</p> <p>Example:CR<<ER*</p> <p>In this example, an item is selected if the first two characters are CR, the third and fourth characters are alpha, and the fifth and sixth characters are ER. Any remaining characters are also allowed.</p>
>	6E	<p>Similar to the percent sign (%) value, the "greater than" character (>) denotes a placeholder value <i>for only numeric characters</i>. The > character can be used in any position order.</p> <p>Example:CR>>ER*</p> <p>In this example, an item is selected if the first two characters are CR, the third and fourth characters are numeric, and the fifth and sixth characters are ER. Any remaining characters are also allowed.</p>
**	5C5C	<p>Using two asterisks (**) allows compatibility with standard ACS and <i>DFSMSdss</i> filtering masks. Only used for <i>partially</i> qualified data set names. The * or ** characters can be used in any qualifier position to denote a <i>wildcard</i> node.</p> <p>Example:CRFM*. *VER.**</p> <p>In this example, a data set entry is selected if it has at least two qualifiers, the first qualifier starts with CRFM, and the second qualifier ends with VER. Any remaining qualifiers are also allowed.</p> <p>Note: As with common data set name masking, any combination of *, **, %, < and/or > characters can be used for the item mask value.</p>

Table 1. Filtering pattern masks (continued)

Mask	Hex	Description
!	5A	Similar to the percent sign (%) value, the exclamation point character (!) denotes a placeholder value <i>for only national characters</i> , based on the U.S.A. EBCDIC code set. Refer to the topic “Use of the U.S.A. EBCDIC code set” on page 22 for more information. The exclamation point (!) can be used in any position order. National characters (based on the U.S.A. EBCDIC code set) are: @ # \$

Worksheets: Gathering required data set names

Identify and record the data set names that will be used during the customization process and make sure that requirements for certain data sets are met.

Data set names for Tools Customizer

Identify and record the following Tools Customizer data set names:

Data set name	Description	Special requirements	Your data set name
SCCQDENU	Metadata library for Tools Customizer		
SCCQLOAD	Executable load module library for Tools Customizer		
SCCQMENU	ISPF messages for Tools Customizer		
SCCQPENU	ISPF panels for Tools Customizer		
SCCQSAMP	Sample members for Tools Customizer		
SCCQTENU	Table library for Tools Customizer	You must have write access to this data set.	

Data set names of DB2 Cloning Tool

Identify and record the following DB2 Cloning Tool data set names. During the customization process, you will enter the following values on panel CCQPPRD.

Data set name	Description	Special requirements	Your data set name
Installation high-level qualifier	The high level qualifier of the DB2 Cloning Tool installed libraries.		
Installation high-level qualifier of previous release of DB2 Cloning Tool	The high level qualifier of the previous release of DB2 Cloning Tool installed libraries.	Only needed if upgrading from a previous release.	

Data set name	Description	Special requirements	Your data set name
SCKZDBRM	DBRM library for DB2 Cloning Tool		
SCKZDENU	Metadata library for DB2 Cloning Tool		
SCKZJCL	Sample members for DB2 Cloning Tool		
SCKZLOAD	Executable load module library for DB2 Cloning Tool	You must APF authorize this data set.	
SCKZMENU	ISPF messages for DB2 Cloning Tool		
SCKZPARM	Parameter library for DB2 Cloning Tool		
SCKZPENU	ISPF panels for DB2 Cloning Tool		
SCKZSLIB	Skeleton library for DB2 Cloning Tool		

Data set names of other libraries

Identify and record the following data set names. During the customization process, you will enter the following values on the Setup panel.

Data set name	Description	Special requirements	Your data set name
Discover output data set	<p>Contains the output that is generated when you run the DB2 Cloning Tool Discover EXEC.</p> <p>The DB2 Cloning ToolDiscover EXEC retrieves the metadata and values for the parameters from a previous customization of DB2 Cloning Tool.</p> <p>The default name of the data set is DB2TOOL.CCQ110.DISCOVER. You can change the default value on the Tools Customizer Settings panel or the Discover Customized Product Information panel.</p>	You must have write access to this data set.	

Data set name	Description	Special requirements	Your data set name
Data store data set	<p>Contains product, LPAR, and DB2 parameter values, and DB2 entry associations. Tools Customizer uses this data set to permanently store all information that is acquired about the product, DB2 subsystems, and LPAR when you customize products on the local LPAR.</p> <p>The default name of the data set is DB2TOOL.CCQ110.DASTOR. You can change the default value on the Tool Customizer Settings panel.</p>	You must have write access to this data set.	
Product customization Library	<p>Contains the customization jobs that Tools Customizer generates for DB2 Cloning Tool.</p> <p>To customize DB2 Cloning Tool, submit the members of the data set in the order in which they are displayed on the Finish Product Customization panel. The data set naming convention is:</p> <p><i>hlq.\$LPAR-name\$.xyzvrm</i></p> <p>where:</p> <ul style="list-style-type: none"> • <i>hlq</i> is the value of the Customization library qualifier field on the Tools Customizer Settings panel (CCQPSET) • <i>LPAR-name</i> is the four-character LPAR name • <i>xyzvrm</i> is the three-letter product identifier with the version, release, and modification level <p>For example, the data set name might be DB2TOOL.PRODUCT.CUST.\$MVS1\$.XYZ410.</p>	You must have write access to this data set.	

APF authorizing load libraries

Some of the programs in DB2 Cloning Tool load libraries must be APF-authorized to run.

About this task

The SCKZLOAD library for DB2 Cloning Tool must have APF authorization. Consult your Systems Programmer to have this library added into the APF list and ensure appropriate access controls have been established

Procedure

Include the SCKZLOAD load library as part of your authorized list.

Note: In general, do not place product modules in LINKLST because doing so could introduce a member name conflict with another product.

Worksheets: Gathering parameter values for Tools Customizer

During the customization process, you must provide parameter values for DB2 Cloning Tool, for DB2, and for your LPAR.

Use the worksheets in this topic to record the appropriate parameter settings for your purposes, and then use these worksheets during the customization process. The worksheets are organized based on the order of the customization panels in the Tools Customizer.

Customization values for the Discover EXEC

Description

Use the following worksheet to identify and record the customization values for the Tools Customizer Discover EXEC. The values in this worksheet are for extracting information from a product that has already been customized. During the customization process, you will enter these values on panel CCQPDSC.

Note: Complete this worksheet only if you are recustomizing a product that has previously been customized by using Tools Customizer.

Parameter	Sample or default value	Your value
Discover EXEC for Extracting Information from an Already Customized product		
Discover EXEC library The fully qualified data set name that contains the product Discover EXEC.	The metadata library that you specified on the Specify the Metadata Library panel (CCQPHLQ).	
Discover EXEC name The name of the Discover EXEC.	CKZDISC	
Discover output data set The name of the data set for the output from the product Discover EXEC.	The name of the discover output library that you entered on the settings panel.	
Information for Discover EXEC section		

Parameter	Sample or default value	Your value
Previous release DB2 Cloning Tool CKZINI library The previous release of the DB2 Cloning Tool CKZINI library data set name.	CKZ220.SCKZPARM	
DB2 Cloning Tool ISPF control data set The data set name of the DB2 Cloning Tool ISPF interface control data set. If no control data set will be used, or you do not want DISCOVER to retrieve DB2 information from a previous control data set, leave this field blank.	blank	

Product to Customize section

Description

The parameters that are listed in the Product to Customize section are read-only. They contain information that was provided on other panels, by Tools Customizer, or by the DB2 Cloning Tool metadata data set.

Parameter	Discovered?	Source of this value
Product metadata library This value is the library that you specified on the Specify the Product to Customize panel. This field is scrollable. Place your cursor anywhere on the field and press PF11 to view its full contents.	No	This value is specified on the Specify the Product to Customize panel (CCQPHLQ)
LPAR The LPAR field displays the LPAR on which you are customizing DB2 Cloning Tool.	No	This value is supplied by Tools Customizer.
Product name This value displays the product that is being customized. In this example, IBM DB2 Cloning Tool should be displayed in this field. This field is scrollable. Place your cursor anywhere on the field and press PF11 to view its full contents.	No	This value is provided by the product metadata file.
Version The Version field displays the version, release and maintenance of the product you are customizing in the format <i>Vn.Rn.mn</i> .	No	This value is provided by the product metadata file.
Product customization library This value displays the name of the data set in which the generated library customization jobs will be stored.	No	This value is derived from the user-specified customization library qualifier on the Tools Customizer Settings panel (CCQPSET).

Required parameters section

Description

The parameters in this task are required for all customizations. During the customization process, you will enter these values on panel CCQPPRD.

Parameter	Required?	Discovered?	Default value	Your value
HLQ of DB2 Cloning Tool installation The high-level qualifier of the DB2 Cloning Tool libraries.	Yes	Yes	CKZ.CKZ310	

Task: Run INIMERGE

Description

This task builds the job to run INIMERGE, which merges a previous CKZINI member or creates a new CKZINI member. The CKZINI member is the product's initialization member, and defines global information regarding DB2 Cloning Tool usage and options within your installation. One job will be generated from this step. During the customization process, you will enter these values on panel CCQPPRD.

This task is *required*.

Jobs generated

This task generates the following job using the listed template:

Table 2. Parameter values

Jobname	Template	Description
ssIMRG	CKZIMRG	Run INIMERGE to merge a previous CKZINI or create a new CKZINI member.

Required authority

The user ID that runs the ssIMRG job must have the following access to DB2 Cloning Tool data sets:

- Read access to the previous release SCKZPARM data set
- Update access to the current release SCKZPARM data set
- Execute access to the current release SCKZLOAD data set

Step or parameter	Required?	Discovered?	Default value	Your value
Run INIMERGE This step builds the job to run INIMERGE.	Yes	No	Selected	
Location of the CKZINI currently in use The location of the CKZINI member that is currently in use. If the CKZINI that is currently in use is in the library for the current release, specify CURRENT-RELEASE. If the location of the CKZINI currently in use is in the previous release, specify PREVIOUS-RELEASE. If a CKZINI is not currently in use, specify NONE.	Yes	No	PREVIOUS-RELEASE	

Step or parameter	Required?	Discovered?	Default value	Your value
HLQ of previous release DB2 Cloning Tool installation The high-level qualifier of the previous release of the DB2 Cloning Tool libraries. This value is ignored if this is a new customization of DB2 Cloning Tool.	Yes	Yes	CKZ.CKZ220	

Task: Subsystem Cloning tasks

Description

This set of tasks binds the Subsystem Cloning plan and packages and defines the Subsystem Cloning stored procedure.

To update the DB2 catalog on the target DB2 subsystem, the DB2 Cloning Tool DB2SQL command has a plan and package that must be installed. If you bind the plan and package on the source DB2 subsystem before cloning, the plan and package will be available for use on the target DB2 subsystem when needed without having to explicitly bind them on the target DB2 subsystem after cloning.

The plan name used by DB2 Cloning Tool Subsystem Cloning must be the same plan name specified by the parameter DB2_PLAN in the CKZINI member, and must also be the same plan name used in the bind. The package name can be any value.

To use the DB2 Cloning Tool Subsystem Cloning stored procedure, it must be defined to DB2 and its package must be bound. The package path value used to define the stored procedure should be the same as the package name used to bind the stored procedure.

This task creates the objects that are required to run Fast Apply on a DB2 subsystem only if those objects were not previously created in this installation or a previous installation. During the customization process, you will enter these values on panel CCQPPRD.

Some of these tasks are *required* and some are *optional*.

Jobs generated

This task generates the following jobs using the listed templates:

Jobname	Template	Description	Required authority to run the job
ssBNDPdd	CKZBNDPS	Subsystem Cloning plan bind	BIND authority on the DB2 system and read access to SCKZDBRM
ssBNDsdd	CKZBNDSS	Subsystem Cloning package bind	BIND authority on the DB2 system and read access to SCKZDBRM
ssDROSdd	CKZDROSP	Drop existing Subsystem Cloning stored procedure	DROP authority on the stored procedure on the DB2 system

Jobname	Template	Description	Required authority to run the job
ssDEFSdd	CKZDEFSP	Define Subsystem Cloning stored procedure	CREATE PROCEDURE authority on the DB2 system
ssBNDSDd	CKZBNDSPP	Bind the Subsystem Cloning stored procedure package	BIND authority on the DB2 system, and read access to SCKZDBRM
ssSSP1	CKZSSP1	Allocate the Subsystem Cloning stored procedure parameters data set	Authority to create the data set
ssSSP2	CKZSSP2	Allocate the Subsystem Cloning stored procedure cloning data sets	Authority to create the data sets
ssSSPP	CKZSSPP	Create the Subsystem Cloning stored procedure product parameters	Authority to update the data set
ssSSPSdd	CKZSSPS	Create the Subsystem Cloning stored procedure DB2 system parameters	Authority to update the data set

Required authority

Refer to the job list for the required authorizations.

Step or parameter	Required?	Discovered?	Default value	Your value
Run Subsystem Cloning plan bind Binds the Subsystem Cloning plan. One job per DB2 entry will be generated.	Yes	No	Selected	
DB2 plan name The DB2 plan name to use for DB2 Cloning Tool. This plan name must be used for both Subsystem Cloning and Table Space Cloning plans.	Yes	Yes	CKZPLAN	
Run Subsystem Cloning package bind Binds the Subsystem Cloning package. One job per DB2 entry will be generated.	Yes	No	Selected	
Drop Subsystem Cloning stored procedure Drops the Subsystem Cloning stored procedure. One job per DB2 entry will be generated. Use this step to drop a previously created stored procedure.	No	No	Not selected	

Step or parameter	Required?	Discovered?	Default value	Your value
Define Subsystem Cloning stored procedure Defines the Subsystem Cloning stored procedure. One job per DB2 entry will be generated. You must run the Subsystem Cloning stored procedure package bind before you can use the Subsystem Cloning stored procedure.	No	No	Not selected	
Run Subsystem Cloning stored procedure package bind Binds the Subsystem Cloning stored procedure package. One job per DB2 entry will be generated. You must run the Subsystem Cloning stored procedure package bind before you can use the Subsystem Cloning stored procedure.	No	No	Not selected	
Allocate Subsystem Cloning stored procedure parameters data set Allocates the Subsystem Cloning stored procedure parameters data set. One job will be generated. The prefix.PARMS data set is created. This data set is used to hold the parameter files used by the stored procedure.	No	No	Not selected	
Stored procedure parms library prefix The prefix for the Subsystem Cloning stored procedure parameters data set.	No	No	CKZ.SP	
Stored procedure data sets VOLSER The volume serial number (VOLSER) for the Subsystem Cloning stored procedure data sets. If the data sets will be managed by SMS, leave this field blank.	No	No	blank	
Stored procedure data sets type The data set type for the Subsystem Cloning stored procedure data sets. For a PDS, specify PDS. For a PDSE, specify LIBRARY.	No	No	LIBRARY	
Allocate Subsystem Cloning stored procedure cloning data sets Allocates the Subsystem Cloning stored procedure cloning data sets. One job will be generated. The prefix.JCL data set and prefix.STATUS data set are created. These data sets are used for a single cloning done by the stored procedure.	No	No	Not selected	

Step or parameter	Required?	Discovered?	Default value	Your value
Stored procedure cloning data sets prefix The prefix for the Subsystem Cloning stored procedure cloning data sets. These data sets will be used by the Subsystem Cloning stored procedure to hold the JCL and status of an individual cloning.	No	No	CKZ.SP.CLONE	
Stored procedure data sets VOLSER The volume serial number (VOLSER) for the Subsystem Cloning stored procedure data sets. If the data sets will be managed by SMS, leave this field blank.	No	No	blank	
Stored procedure data sets type The data set type for the Subsystem Cloning stored procedure data sets. For a PDS, specify PDS. For a PDSE, specify LIBRARY.	No	No	LIBRARY	
Create Subsystem Cloning stored procedure product parameters Create the Subsystem Cloning stored procedure product parameters member in the parameters library. One job will be generated. This job will create the PPARM member in the prefix.PARMS data set.	No	No	Not selected	
Stored procedure parms library prefix The prefix for the Subsystem Cloning stored procedure parameters data set.	No	No	CKZ.SP	
Create Subsystem Cloning stored procedure system parameters Create the Subsystem Cloning stored procedure system parameters member in the parameters library. One job per DB2 entry will be generated. This job adds DB2 system parameters for the selected DB2 entry to the SPARM member of the prefix.PARMS data set.	No	No	Not selected	
Stored procedure parms library prefix The prefix for the Subsystem Cloning stored procedure parameters data set.	No	No	CKZ.SP	

Task: Table Space Cloning tasks

Description

This set of tasks binds the Table Space Cloning plan and packages and allocates the Table Space Cloning runtime repository data sets.

DB2 Cloning Tool Table Space Cloning uses static and dynamic SQL to access the catalog and issue DB2 ALTER commands. If you are planning on using Table Space Cloning, a bind is required on each subsystem on which Table Space Cloning will execute. This includes all three Table Space Cloning execution types: source, target, and TCP/IP server.

The plan name used by DB2 Cloning Tool Table Space Cloning must be the same plan name specified by the parameter DB2_PLAN in the CKZINI member, and must also be the same plan name used in the bind. The package names can be any value.

During the customization process, you will enter these values on panel CCQPPRD.

Some of these tasks are *required* and some are *optional*.

Jobs generated

This task generates the following jobs using the listed templates:

Table 3. Parameter values

Jobname	Template	Description	Required authority to run the job
ssBNDPdd	CKZBNDPT	Table Space Cloning plan bind	BIND authority on the DB2 system, and read access to SCKZDBRM
ssBNDTdd	CKZBNPTS	Table Space Cloning package binds	BIND authority on the DB2 system, and read access to SCKZDBRM
ssBNDTdd	CKZBNDTD	Table Space Cloning package binds for DDF location	BIND authority on the DB2 system, and read access to SCKZDBRM
ssTS1	CKZTS1	Allocate Table Space Cloning runtime repository data sets	Authority to create the data sets
ssTS2	CKZTS2	Allocate Table Space Cloning data sets	Authority to create the data sets
ssTS3	CKZTS3	Create XML create DDL member	Authority to update the data set prefix.XMLCRDDL
ssDROLdd	CKZDROLA	Drop log apply table	Authority to DROP the log apply table and associated database, table space, and indexes on the DB2 system
ssDEFLdd	CKZDEFLA	Create log apply table	Authority to CREATE the log apply table and associated database, table space, and indexes on the DB2 system

Table 3. Parameter values (continued)

Jobname	Template	Description	Required authority to run the job
ssALTLdd	CKZALTLA	Upgrade log apply table for DB2 V11 support	Authority to ALTER DROP the log apply table on the DB2 system

Required authority

Refer to the job list for the required authorizations.

Step or parameter	Required?	Discovered?	Default value	Your value
Run Table Space Cloning plan bind Binds the Table Space Cloning plan. One job per DB2 entry will be generated.	Yes	No	Selected	
DB2 plan name The DB2 plan name to use for DB2 Cloning Tool Table Space Cloning. This plan name must be used for both Subsystem Cloning and Table Space Cloning plans.	Yes	Yes	CKZPLAN	
Run Table Space Cloning package binds Binds the Table Space Cloning packages. One job per DB2 entry will be generated.	Yes	No	Selected	
Run Table Space Cloning package binds for DDF location Binds the Table Space Cloning packages for the DDF location. One job per DB2 entry will be generated. If a DDF location is not defined for the selected DB2 entry, a null job is generated.	No	No	Not selected	
Allocate Table Space Cloning runtime repository data sets Allocates the Table Space Cloning runtime repository data sets. One job will be generated. The prefix.RRJOB data set and prefix.RRDSN data set are created.	No	No	Not selected	
Table Space Cloning runtime repository data sets prefix The prefix for the Table Space Cloning runtime repository data sets.	No	No	CKZ.TSREPOS	

Step or parameter	Required?	Discovered?	Default value	Your value
Table Space Cloning runtime repository data sets VOLSER The volume serial number (VOLSER) to be used for the Table Space Cloning repository data sets. If data sets will be managed by SMS, leave this field blank.	No	No	blank	
Allocate Table Space Cloning data sets Allocates the Table Space Cloning data sets. One job will be generated. The following data sets are created: prefix.CKZIN, prefix.CMDSSTPT, prefix.CMDSSTPS, prefix.CMDSSTRS, prefix.COPYDSNS, prefix.IDCAMS, prefix.LISTDEF, prefix.LOGAPCTL, prefix.MASKDEF, prefix.RDREPL, prefix.RDREPS, prefix.RRJREPL, prefix.RRJREPS, prefix.SQLOUT, prefix.SYNCDB2, prefix.TDDIN, prefix.TDDOUT, prefix.XMLCRDDL, and prefix.XMLSTR.	No	No	Not selected	
Table Space Cloning data sets prefix The prefix for the Table Space Cloning data sets.	No	No	CKZ.TSDS	
Table Space Cloning data sets VOLSER The volume serial number (VOLSER) for the Table Space Cloning data sets. If the data sets will be managed by SMS, leave this field blank.	No	No	blank	
Table Space Cloning data sets type The data set type for the Table Space Cloning data sets. For a PDS, specify PDS. For a PDSE, specify LIBRARY.	No	No	LIBRARY	
Create Table Space Cloning XML create DDL member Creates the XLM create DDL member. The member will be placed in the prefix.XMLCRDDL data set. This DDL is used by Table Space Cloning to create a table that contains an XML column. One job will be generated.	No	No	Not selected	
Table Space Cloning data sets prefix The prefix for the Table Space Cloning data sets.	No	No	CKZ.TSDS	

Step or parameter	Required?	Discovered?	Default value	Your value
Member name for XML create DDL The member name that the XML create DDL will be saved in. The member will be placed in the data set prefix.XMLCRDDL. This DDL is used by Table Space Cloning to create a table that contains an XML column.	No	No	CKZXMDDL	
STOGROUP name for XML create DDL The name of the storage group (STOGROUP) for the XML create DDL. This DDL is used by Table Space Cloning to create a table that contains an XML column.	No	No	SYSDEFLT	
Table creator name for XML create DDL The table creator name for the XML create DDL. This DDL is used by Table Space Cloning to create a table that contains an XML column.	No	No	CKZ	
Drop Table Space Cloning log apply table Drops the log apply table and associated database, table space, and indexes. One job per DB2 entry will be generated.	No	No	Not selected	
Create Table Space Cloning log apply table Creates the log apply table and associated database, table space, and indexes. One job per DB2 entry will be generated.	No	No	Not selected	
Upgrade Table Space Cloning log apply table for DB2 V11 support Upgrades the log apply table for DB2 V11 support. One job per DB2 entry will be generated.	No	No	Not selected	

Task: ISPF interface tasks

Description

This set of tasks creates the ISPF interface CLISTs and allocates data sets that are used by the ISPF interface. During the customization process, you will enter these values on panel CCQPPRD.

These tasks are *required* if you are planning to use the ISPF interface.

Jobs generated

This task generates the following jobs using the listed templates:

Table 4. Parameter values

Jobname	Template	Description	Required authority to run the job
ssISPF1	CKZISPF1	Create the CLISTS that are used to invoke the ISPF interface	Authority to update the CLIST library data set
ssISPF2	CKZISPF2	Allocate ISPF repository data sets	Authority to create the data sets
ssISPF3	CKZISPF3	Allocate ISPF control data set	Authority to create the data sets

Required authority

Refer to the job list for the required authorizations.

Step or parameter	Required?	Discovered?	Default value	Your value
ISPF control data set prefix The prefix for the ISPF interface control data set.	If using ISPF interface	Yes	CKZ.ISPFCONT	
ISPF repository data sets prefix The prefix for the ISPF interface repository data sets.	If using ISPF interface	No	CKZ.ISPFREPO	
Create ISPF interface CLISTS Creates the CLISTS that are used to invoke the ISPF interface. One job will be generated.	If using ISPF interface	No	Not selected	
ISPF CLIST library data set name The data set name of the CLIST library where the ISPF interface CLISTS are to be placed. The CLIST library must already exist and must be a PDS or PDSE that is fixed block (RECFM=FB) and has a record length of 80 (LRECL=80).	If using ISPF interface	No	CLIST.LIBRARY	
Allocate ISPF repository data sets Allocates the necessary repository data sets to be used by the ISPF interface. One job will be generated. The prefix.PROFILES data set and prefix.PROFILE.RPT data set are created.	No	No	Not selected	
ISPF repository data sets VOLSER The volume serial number (VOLSER) to be used for the ISPF interface repository data sets. If the data sets will be managed by SMS, leave this field blank.	No	No	blank	

Step or parameter	Required?	Discovered?	Default value	Your value
Allocate ISPF control data set Allocates the control data set to be used by the ISPF interface. One job will be generated. The prefix.CONTROL data set is created.	No	No	Not selected	
ISPF control data set VOLSER The volume serial number (VOLSER) to be used for the ISPF interface control data sets. If the data sets will be managed by SMS, leave this field blank.	No	No	blank	

DB2 Parameters section

Description

This section contains DB2 parameters. All parameters are required. During the customization process, you will enter these values on panel CCQPDB2.

Parameter	Required?	Discovered?	Default value	Your value
DB2 subsystem ID description A description for the DB2 subsystem. The value must be 72 characters or less.	No	Yes	blank	
General DB2 Information				
Location name The DB2 location name. The value must be 16 characters or less.	Yes	Yes	LOC1	
Mode The mode in which the DB2 subsystem is running. The following values are valid: <ul style="list-style-type: none"> • CM is compatibility mode on all listed DB2 versions except DB2 10. • CM8 is conversion mode from DB2 V8 on DB2 10. • CM9 is conversion mode from DB2 Version 9.1 on DB2 10. • NFM is new function mode on all listed DB2 versions. 	Yes	No	NFM	

Parameter	Required?	Discovered?	Default value	Your value
Level number This parameter indicates the version, release, and modification level of the DB2 subsystem. The following values are valid: <ul style="list-style-type: none"> • 810 is valid only for CM or NFM. • 910 is valid only for CM or NFM. • 101 is valid only for CM8, CM9 or NFM. • 111 is valid only for CM or NFM. 	Yes	No	blank	
DB2 Libraries				
Load library This parameter indicates the data set name of the DB2 load library.	Yes	Yes (see note)	DSN.SDSNLOAD	
Run library This parameter indicates the data set name of the DB2 run library.	Yes	Yes (see note)	DSN.RUNLIB.LOAD	
Exit library This parameter indicates the data set name of the DB2 exit library.	Yes	Yes (see note)	DSN.SDSNEXIT	
Bootstrap data set This parameter indicates the name of the DB2 bootstrap data set.	Yes	Yes	DSN.SDSNBSDS	
DB2 Utilities				
Plan name for the DSNTIAD utility This parameter indicates the plan name for the DSNTIAD utility. The value must be 8 characters or less.	Yes	No	DSNTIAD	
DB2 Cloning Tool parameters				
DB2 package name The DB2 package name to use for DB2 Cloning Tool. This package name is used for both Subsystem Cloning and Table Space Cloning packages.	Yes	No	CKZPACK	
Target DB2 DDF location name for Table Space Cloning The target DB2 DDF location name that the Table Space Cloning source job will connect to by using DDF. If DDF will not be used, leave this field blank.	No	No	blank	
DB2 SQLID for object creates The SQLID to use when creating objects.	Yes	No	CKZSQLID	

Parameter	Required?	Discovered?	Default value	Your value
Stored procedure schema name The schema name for the Subsystem Cloning stored procedure.	Yes	No	CKZTOOLS	
Stored procedure WLM environment name The WLM environment name for the Subsystem Cloning stored procedure. To use the DB2 system default WLM environment, leave this field blank.	No	No	blank	
Table creator name for log apply table The creator name for the Table Space Cloning log apply table.	Yes	No	CKZ	
Database name for log apply table The database name for the Table Space Cloning log apply table.	Yes	No	CKZLADB	
Buffer pool for log apply table The buffer pool name for the Table Space Cloning log apply table.	Yes	No	BP0	
STOGROUP for log apply table The name of the storage group for the Table Space Cloning log apply table.	Yes	No	SYSDEFLT	
DB2 Cloning Tool Subsystem Cloning stored procedure parameters				
System ID where this DB2 normally runs The JES system identifier where this DB2 subsystem normally runs.	No	Yes	blank	
DB2 VSAM catalog name The DB2 VSAM catalog name. This is the high level qualifier used for the DB2 directory and DB2 catalog data sets.	No	Yes	blank	
Special DSNZPARM name The special DSNZPARM name. This is the name of the DSNZPARM to be used when starting the DB2 subsystem in special mode.	No	Yes	blank	
Normal DSNZPARM name The normal DSNZPARM name. This is the name of the DSNZPARM to be used when starting the DB2 subsystem in normal mode.	No	Yes	blank	
Data sharing group name The data sharing group name. This is the name of the data sharing group that this DB2 subsystem is a member of. If not part of a data sharing group, leave this field blank.	No	Yes	blank	

Parameter	Required?	Discovered?	Default value	Your value
Data sharing member name The data sharing member name. This is the member name of this DB2 subsystem in the data sharing group. If not part of a data sharing group, leave this field blank.	No	Yes	blank	
DDF generic LUNAME The DDF generic LUNAME. This parameter corresponds to the GENERIC keyword of the DB2UPDATE command.	No	Yes	blank	
DDF LUNAME The DDF LUNAME. This parameter corresponds to the LUNAME keyword of the DB2UPDATE command.	No	Yes	blank	
DDF password The DDF password. This parameter corresponds to the PASSWORD keyword of the DB2UPDATE command.	No	Yes	blank	
DDF TCP/IP port number The DDF TCP/IP port number. This parameter corresponds to the PORT keyword of the DB2UPDATE command.	No	Yes	blank	
DDF TCP/IP resynchronization port number The DDF TCP/IP resynchronization port number. This parameter corresponds to the RESPORT keyword of the DB2UPDATE command.	No	Yes	blank	
DDF TCP/IP secure port number The DDF TCP/IP secure port number. This parameter corresponds to the SECPORT keyword of the DB2UPDATE command.	No	Yes	blank	
DDF TCP/IP IP name The DDF TCP/IP IP name. This parameter corresponds to the IPNAME keyword of the DB2UPDATE command.	No	Yes	blank	
DDF TCP/IP IPV4 IP address The DDF TCP/IP IPV4 IP address. This parameter corresponds to the IPV4 keyword of the DB2UPDATE command.	No	Yes	blank	

Parameter	Required?	Discovered?	Default value	Your value
DDF TCP/IP IPV6 IP address The DDF TCP/IP IPV6 IP address. This parameter corresponds to the IPV6 keyword of the DB2UPDATE command.	No	Yes	blank	
DDF TCP/IP group IPV4 IP address The DDF TCP/IP group IPV4 IP address. This parameter corresponds to the GRPIPV4 keyword of the DB2UPDATE command.	No	Yes	blank	
DDF TCP/IP group IPV6 IP address The DDF TCP/IP group IPV6 IP address. This parameter corresponds to the GRPIPV6 keyword of the DB2UPDATE command.	No	Yes	blank	
DDF alias name The DDF alias name. This parameter corresponds to the ALIAS keyword of the DB2UPDATE command.	No	Yes	blank	

LPAR Parameters section

Description

DB2 Cloning Tool does not require any LPAR parameters.

Chapter 3. Starting and preparing Tools Customizer for use

Use the provided REXX EXEC to start Tools Customizer. The first time that you use Tools Customizer, you must modify the settings that Tools Customizer uses to customize DB2 Cloning Tool.

Starting Tools Customizer

Start Tools Customizer by running a REXX EXEC from the ISPF Command Shell panel.

Before you begin

Tools Customizer must be SMP/E installed. You must know the high-level qualifier of where the Tools Customizer libraries reside. The high-level qualifier is considered to be all the segments of the data set name except the lowest-level qualifier, which is SCCQEXEC.

Attention: Ensure that Tools Customizer load libraries are not APF authorized. APF authorizing Tools Customizer libraries results in an abend.

About this task

To run the REXX EXEC, you must either change the placeholder in the EXEC for the high-level qualifier of the Tools Customizer EXEC library or pass the high-level qualifier as a parameter when you run the EXEC. The REXX EXEC is in the CCQTCZ member of the EXEC library.

Procedure

1. Optional: Change the placeholder for the high-level qualifier in the REXX EXEC:
 - a. Find the EXEC library data set for Tools Customizer. The name of the data set is *high_level_qualifier.SCCQEXEC*.
 - b. Edit data set member CCQTCZ and replace the <TCZ HLQ> string with the high-level qualifier of the EXEC library data set. For example, if the name of the Tools Customizer EXEC library is CCQTCZ.USABSAND.SCCQEXEC, replace <TCZ HLQ> with CCQTCZ.USABSAND.

You have to change the placeholder for the high-level qualifier only once. When you run the REXX EXEC, you do not have to pass the high-level qualifier as a parameter.

2. Run the REXX EXEC (CCQTCZ):
 - a. From the ISPF Primary Option Menu, select option 6. The ISPF Command Shell panel is displayed.
 - b. Specify the EX command to run the REXX EXEC. For example, if the Tools Customizer EXEC library is CCQTCZ.USABSAND.SCCQEXEC and you changed the placeholder for the high-level qualifier in the REXX EXEC, specify: EX 'CCQTCZ.USABSAND.SCCQEXEC(CCQTCZ)'

If you did not change the placeholder for the high-level qualifier in the REXX EXEC, specify: EX 'CCQTCZ.USABSAND.SCCQEXEC(CCQTCZ)'
'CCQTCZ.USABSAND'

Results

The IBM Customizer Tools for z/OS main menu panel is displayed.

What to do next

If you are running Tools Customizer for the first time, you must modify the Tools Customizer user settings. If you have already set the Tools Customizer user settings, either customize or recustomize DB2 Cloning Tool.

Modifying Tools Customizer user settings

Before you can customize DB2 Cloning Tool with Tools Customizer, you must review the settings that Tools Customizer uses. You might have to change the default values to suit your environment. In most cases, you can change the Tools Customizer values at any time. For example, after you have customized DB2 Cloning Tool and are customizing a different product or solution pack, you might have to change the settings.

Procedure

1. On the IBM Tools Customizer for z/OS main panel (CCQPHME), specify option 0, **User settings for Tools Customizer**. The Tools Customizer Settings panel (CCQPSET) is displayed, as shown in the following figure:

```
CCQPSET          Tools Customizer Settings          14:03:51
Command ==>>>
Enter the settings for customizing a product or press End to save and exit.

Commands: SAVE - Save user settings

Product Customization Settings
Customization library qualifier . . DB2TOOL.PRODUCT.CUST
Use DB2 group attach . . . . . YES (YES/NO)

Tools Customizer Library Settings
Metadata library . . . . . DB2TOOL.CCQ110.SCCQDENU
Discover output data set . DB2TOOL.CCQ110.DISCOVER
Data store data set . . . DB2TOOL.CCQ110.DATASTOR

User Job Card Settings for Customization Jobs
==>> //          JOB
==>>
==>>
==>>
==>>
```

Figure 1. The Tools Customizer Settings panel (CCQPSET)

2. Review the values for the following required fields. Use the default value or specify your own value. You must have appropriate read and write access to the data sets that are specified.

Customization library qualifier

The high-level qualifier that is used as the prefix for the customization library. The customization library is a data set in which the generated jobs to customize DB2 Cloning Tool are stored. Write access to this qualifier is required.

For each product to be customized, the first value that is specified for the qualifier is always used, even if you change it after you have generated the customization jobs. For example, if you customize a

product and then specify a new qualifier for recustomization, although the new qualifier is saved and displayed, the original value is used.

To maintain multiple instances of Tools Customizer, specify a unique customization library qualifier for each instance of Tools Customizer. Data set names that exceed 42 characters must be enclosed in single quotation marks (').

Use DB2 group attach

Determines the value that is used in the CONNECT statements in the generated customization jobs. Specify YES for data sharing environments, which causes the group attach name to be used. Specifying NO, in most cases, causes the SSID to be used in the DB2 CONNECT statement.

Important: This field has no effect when you are customizing a product on a DB2 subsystem that is not a member of a data sharing group. In this case, the DB2 subsystem ID (SSID) is always used in the CONNECT statements in the generated customization jobs.

When you are customizing a product on a DB2 subsystem that is a member of a data sharing group, how the DB2 subsystem is defined and the value of the **Use DB2 group attach** field determines the value that is used in the CONNECT statements in the generated jobs. The following table shows whether the SSID or the group attach name is used:

Table 5. The effect of the value of the Use DB2 group attach field in a data sharing environment

DB2 subsystem definition	Value of the Use DB2 group attach field	Value that is used in the CONNECT statements
The DB2 subsystem is defined with an SSID.	Yes	Group attach name
	No	SSID ¹
The DB2 subsystem is not defined with an SSID.	Yes or No	Group attach name

Note 1: If you generate jobs for multiple DB2 subsystems that are defined with an SSID and belong to the same data sharing group, the SSID of the first DB2 subsystem that is selected is used.

For example, assume that on the Customizer Workplace panel, you generated jobs for the following DB2 subsystems:

- V91C, which is a stand-alone DB2 subsystem
- V91A, which is a DB2 subsystem that is a member of data sharing group DSG1
- A DB2 subsystem that was not defined with an SSID that is a member of data sharing group DSGA

The following figure shows how these DB2 entries might be listed on the Customizer Workplace panel:

```

Associated DB2 Entries and Parameter Status
Line commands: G - Generate jobs E - Edit B - Browse C - Copy R - Remove
Cmd SSID GrpAttach Lvl Mode User ID Date Status Message
V91C -- 910 NFM SYSADM 2010/11/09 Ready to Customize
V91A DSG1 910 NFM SYSADM 2010/11/09 Ready to Customize
-- DSGA 910 NFM SYSADM 2010/11/09 Ready to Customize
----- End of DB2 entries -----

```

The following table shows which values are used in the CONNECT statements in the generated jobs, based on the value of the **Use DB2 group attach** field.

Table 6. Value that is used in the CONNECT statements in the generated jobs

SSID	GrpAttach	Value of the Use DB2 group attach field	Value that is used in the CONNECT statements
V91C	--	Yes	SSID
		No	SSID
V91A	DSG1	Yes	Group attach name
		No	SSID
--	DSGA	Yes	Group attach name
		No	Group attach name

Tools Customizer metadata library

The name of the data set that contains the metadata that is used to display the DB2 parameters. The parameters that are displayed on the DB2 Parameters panel depend on the parameters that you define and the tasks and steps that you select on the Product Parameters panel for the product that you are customizing. For example, the DB2 parameters that are required, based on the selected tasks and steps, are displayed on the DB2 Parameters panel, and you can edit them. If they are not required, they are not displayed. Read access to this data set is required. Data set names that exceed 42 characters must be enclosed in single quotation marks (').

Discover output data set

The name of the data set in which the output from the DB2 Cloning Tool Discover EXEC is stored. Each product has its own Discover EXEC. The Discover EXEC retrieves the product and DB2 parameters from a previously customized product. Write access to this data set is required. Data set names that exceed 42 characters must be enclosed in single quotation marks (').

Data store data set

The name of the data set where Tools Customizer stores information about product and DB2 parameter values. Information about which products are associated with which DB2 entries (DB2 subsystems, DB2 group attach names, and DB2 data sharing members) is also stored in this data set. Data set names that exceed 42 characters must be enclosed in single quotation marks ('). The specified data store data set can be used with only one invocation of Tools Customizer at a time. Data set names that exceed 42 characters must be enclosed in single quotation marks (').

User job card settings for customization jobs

The job card information to be inserted into the generated jobs for

customizing a product. The default value is the job statement information from the ISPF Batch Selection panel.

The first line of the job card automatically begins with the following information:

```
//          JOB
```

where characters 3 - 10 are reserved by Tools Customizer for the job name and includes a blank space after JOB. This name cannot be edited. Information that you specify on the first line of the job card cannot exceed 57 characters. This character limit includes a continuation character. All other lines of the job card cannot exceed 72 characters.

3. Press End to save and exit. If the Discover output data set and the data store data set that you specified do not exist, Tools Customizer creates them.

Important: If the ISPF sessions unexpectedly ends before you exit Tools Customizer, the fields on the Tools Customizer Settings panel (CCQPSET) will be repopulated with default values, and you will be required to review them or specify new values again.

Results

The values are saved, and the IBM Tools Customizer for z/OS main menu panel (CCQPHME) is displayed again.

What to do next

You are ready to customize or recustomize DB2 Cloning Tool or to change parameter settings.

Chapter 4. Customizing DB2 Cloning Tool

Using Tools Customizer to customize DB2 Cloning Tool consists of identifying the product to customize; defining any required DB2 Cloning Tool and DB2 parameters; generating the customization jobs; and submitting the jobs.

Customization roadmaps describe the steps that you must complete to customize DB2 Cloning Tool. Separate roadmaps are provided for the three most common types of customizations.

Use the following table to determine which roadmap corresponds to your environment.

Table 7. Customization roadmaps

Environment description	Roadmap
You do not have a customized version of DB2 Cloning Tool, and you need to customize it for the first time.	"Roadmap: Customizing DB2 Cloning Tool for the first time"
You have already customized a version of DB2 Cloning Tool, and you want to use the same parameter values to customize a different version.	"Roadmap: Customizing a new version of DB2 Cloning Tool from a previous customization" on page 52
You have a customized version of of DB2 Cloning Tool, but you want to change one or more parameter values.	"Roadmap: Recustomizing DB2 Cloning Tool" on page 53

Roadmap: Customizing DB2 Cloning Tool for the first time

This roadmap lists and describes the steps that are required to customize DB2 Cloning Tool for the first time.

If you are customizing a previous version of DB2 Cloning Tool, see "Roadmap: Customizing a new version of DB2 Cloning Tool from a previous customization" on page 52.

Before you complete these steps, ensure that the following prerequisites have been met:

- All of the product customization steps that must be done before Tools Customizer is started are complete.
- Tools Customizer is started.
- The Tools Customizer settings have been reviewed or modified, and saved.

Complete the steps in the following table to customize DB2 Cloning Tool for the first time.

Table 8. Steps for customizing DB2 Cloning Tool for the first time

Step	Description	Instructions
1	Specify the product metadata library for the product that you want to customize. The name of this library is <i>hlq.SCKZDENU</i> .	"Specifying the metadata library for the product to customize" on page 54
2	Create new DB2 entries and associate them with DB2 Cloning Tool.	"Creating and associating DB2 entries" on page 58
3	Define the required parameters.	"Defining parameters" on page 59
4	Generate the customization jobs for the product or for the DB2 entries on which DB2 Cloning Tool is ready to be customized.	"Generating customization jobs" on page 63
5	Submit the generated customization jobs.	"Submitting customization jobs" on page 63

The following table lists some of the common administrative tasks that you might need to do during the customization process.

Table 9. Administrative tasks

Description	Instructions
Browse the different types of parameters.	"Browsing parameters" on page 65
Copy an existing DB2 entry to the list of DB2 entries on which DB2 Cloning Tool can be customized.	"Copying DB2 entries" on page 66
Remove one or more DB2 entries from the associated list.	"Removing DB2 entries" on page 67
Delete one or more DB2 entries from the master list.	"Deleting DB2 entries" on page 68
Display a list of customization jobs that have been previously generated.	"Displaying customization jobs" on page 68
Maintain the customization jobs in the customization library.	"Maintaining customization jobs" on page 68

Roadmap: Customizing a new version of DB2 Cloning Tool from a previous customization

This roadmap lists and describes the steps for customizing a new version of DB2 Cloning Tool based on the existing customization values of a previous version of the same product.

Use this roadmap even if the previous version of DB2 Cloning Tool was not customized by using Tools Customizer.

Before you complete these steps, ensure that the following prerequisites have been met:

- All of the product customization steps that must be done before Tools Customizer is started are complete.
- Tools Customizer is started.
- The Tools Customizer settings have been reviewed or modified, and saved.

Complete the steps in the following table to customize a new version of DB2 Cloning Tool from a previous customization.

Table 10. Steps for customizing a new version of DB2 Cloning Tool from a previous customization

Step	Description	Instructions
1	Specify the product metadata library for the product that you want to customize. The name of this library is <i>hlq.SCKZDENU</i> .	“Specifying the metadata library for the product to customize” on page 54
2	Use the DB2 Cloning Tool Discover EXEC to discover information about the version of DB2 Cloning Tool that you previously customized manually.	“Discovering DB2 Cloning Tool information automatically” on page 56
3	Define the required parameters.	“Defining parameters” on page 59
4	Generate the customization jobs for the product or for the DB2 entries on which DB2 Cloning Tool is ready to be customized.	“Generating customization jobs” on page 63
5	Submit the generated customization jobs.	“Submitting customization jobs” on page 63

The following table lists some of the common administrative tasks that you might need to do during the customization process.

Table 11. Administrative tasks

Description	Instructions
Browse the different types of parameters.	“Browsing parameters” on page 65
Copy an existing DB2 entry to the list of DB2 entries on which DB2 Cloning Tool can be customized.	“Copying DB2 entries” on page 66
Remove one or more DB2 entries from the associated list.	“Removing DB2 entries” on page 67
Delete one or more DB2 entries from the master list.	“Deleting DB2 entries” on page 68
Display a list of customization jobs that have been previously generated.	“Displaying customization jobs” on page 68
Maintain the customization jobs in the customization library.	“Maintaining customization jobs” on page 68

Roadmap: Recustomizing DB2 Cloning Tool

This roadmap lists and describes the steps to change parameter values and regenerate customization jobs for DB2 Cloning Tool after you have customized it for the first time.

The new customization jobs will replace the customization jobs that were previously generated and stored in the customization library. Part of the recustomization process includes selecting or deselecting optional tasks or steps, changing the definitions of parameters that have already been defined, or both. Use the method in this roadmap instead of deleting customization jobs from the customization library.

Before you complete these steps, ensure that the following prerequisites have been met:

- All of the product customization steps that must be done before Tools Customizer is started are complete.
- Tools Customizer is started.

Complete the steps in the following table to recustomize DB2 Cloning Tool.

Table 12. Required steps for recustomizing DB2 Cloning Tool

Step	Description	Instructions
1	Specify the product metadata library for the product that you want to recustomize. The name of this library is <i>hlq.SCKZDENU</i> .	"Specifying the metadata library for the product to customize"
2	Edit the specific tasks, steps, or parameters that need to be changed.	<ul style="list-style-type: none"> • "Defining DB2 Cloning Tool parameters" on page 60 • "Defining DB2 parameters" on page 61
3	Generate the customization jobs for the product or for the DB2 entries on which DB2 Cloning Tool is ready to be customized.	"Generating customization jobs" on page 63
4	Submit the new generated customization jobs.	"Submitting customization jobs" on page 63

The following table lists some of the common administrative tasks that you might need to do during the customization process.

Table 13. Administrative tasks

Description	Instructions
Browse the different types of parameters.	"Browsing parameters" on page 65
Copy an existing DB2 entry to the list of DB2 entries on which DB2 Cloning Tool can be customized.	"Copying DB2 entries" on page 66
Remove one or more DB2 entries from the associated list.	"Removing DB2 entries" on page 67
Delete one or more DB2 entries from the master list.	"Deleting DB2 entries" on page 68
Display a list of customization jobs that have been previously generated.	"Displaying customization jobs" on page 68
Maintain the customization jobs in the customization library.	"Maintaining customization jobs" on page 68

Specifying the metadata library for the product to customize

You must specify a metadata library for the product that you want to customize.

About this task

The product metadata library contains the information that determines which tasks, steps, and parameters are required to customize DB2 Cloning Tool. This information controls what is displayed on the Product Parameters panel and the DB2 Parameters panel.

After DB2 Cloning Tool has been SMP/E installed, the default name of the product metadata library is *high_level_qualifier.SCKZDENU*, where *high_level_qualifier* is all of the segments of the data set name except the lowest-level qualifier.

Procedure

1. Specify option 1 on the Tools Customizer for z/OS panel. The Specify the Metadata Library panel is displayed. This panel contains a list of the product metadata libraries that you specified most recently. If you are using Tools Customizer for the first time, this list is empty, as shown in the following figure:

```
CCQPHLQ          Specify the Metadata Library          16:07:45
Command ==>>>          Scroll ==>> CSR

Type the name of the metadata library for the pack or the product in the
Metadata library field, or select the library in the list of previous
libraries and press Enter to populate the field. Press Enter to continue.

The default name of the metadata library after the pack or product has been
SMP/E installed is <hlq>.SxxxDENU, where <hlq> is the high-level qualifier for
the pack or the product, and xxx is the 3-character prefix for the pack or
the product.

Metadata library . CKZ.V310.SCKZDENU

Previously Used Metadata Library:

=>
=>
=>
=>
```

Figure 2. The Specify the Metadata Library panel

2. Use one of the following methods to specify the product metadata library:
 - Type the name of a fully qualified partitioned data set (PDS) or an extended partitioned data set (PDSE) in the **Metadata library** field. Double quotation marks (") cannot be used around the name. Single quotation marks (') can be used but are not required. If you are customizing DB2 Cloning Tool for the first time, you must use this method.
 - Place the cursor on the library name in the Recent Metadata Libraries list, and press Enter.

Results

If you are customizing DB2 Cloning Tool for the first time, the Run Discover EXEC panel is displayed. Otherwise, the Customizer Workplace panel is displayed.

What to do next

- Complete the steps that correspond to your environment:

Customizing DB2 Cloning Tool for the first time

Do not run the DB2 Cloning Tool Discover EXEC. Press End. The

Customizer Workplace panel is displayed. If your environment requires associated DB2 entries, ensure that they are created and associated. If your environment does not require associated DB2 entries, skip this step, and edit DB2 Cloning Tool parameters.

Customizing DB2 Cloning Tool from a previous or current customization

Press Enter to run the DB2 Cloning Tool Discover EXEC. The Discover Customized Product Information panel is displayed. Specify the required information for running the EXEC.

Discovering DB2 Cloning Tool information automatically

You can use the DB2 Cloning Tool Discover EXEC to discover information from a previous or current customization of DB2 Cloning Tool.

About this task

Tip: Using the DB2 Cloning Tool Discover EXEC to discover information from a previous or current customization saves time and reduces errors that can occur when parameters are specified manually.

DB2 Cloning Tool provides the Discover EXEC that you will run. Therefore, the information that can be discovered depends on DB2 Cloning Tool.

Parameter values that are discovered and parameter values that are specified manually are saved in the data store. If parameter values for the product that you want to customize exist in the data store, Tools Customizer issues a warning before existing values are replaced.

Procedure

1. On the Customizer Workplace panel, issue the DISCOVER command. If you chose to run the DB2 Cloning Tool Discover EXEC on the pop-up panel after you specified the product to customize, skip this step.

Tip: You can run any Tools Customizer primary command by using either of the following methods:

- Place the cursor on the name of the primary command, and press Enter.
- Type the primary command name in the command line, and press Enter.

The Discover Customized Product Information panel is displayed, as shown in the following figure:

```

CCQPDSC          Discover Customized Product Information          16:09:40
Command ==>>>                               Scroll ==>> CSR

For the product you are customizing, the Discover EXEC retrieves product
information from an already customized product. Specify the required
information, and press Enter to run the Discover EXEC. Press End to cancel.

Product to Customize
Product metadata library : CKZ.V310.SCKZDENU          > LPAR. . . : RS22
Product name . . . . . : DB2 Cloning Tool for z/OS > Version . . : 3.1.0

Discover EXEC for Extracting Information from an Already Customized Product
Discover EXEC library . . . CKZ.V310.SCKZDENU
Discover EXEC name . . . . . CKZDISC
Discover output data set . . TWUSR.TCZ.DISCOVER

Information for Discover EXEC
Previous release DB2 Cloning Tool CKZINI library
                                CKZ220.SCKZPARM          >

```

Figure 3. The Discover Customized Product Information panel

2. Either accept the default values for the following input fields that Tools Customizer generates, or replace the default values with your own values:

Discover EXEC library

The fully qualified data set name that contains the DB2 Cloning Tool Discover EXEC.

Discover EXEC name

The name of the DB2 Cloning Tool Discover EXEC.

Discover output data set

The fully qualified data set where output from the DB2 Cloning Tool Discover EXEC is stored.

3. Either accept or change the default values in the **Information for Discover EXEC** fields. These fields are generated by DB2 Cloning Tool. They show the information that is required to run the DB2 Cloning Tool Discover EXEC.
4. Issue the RUN command to run the DB2 Cloning Tool Discover EXEC. Alternatively, save your information without running the DB2 Cloning Tool Discover EXEC by issuing the SAVE command. If you issue the RUN command to run the DB2 Cloning Tool Discover EXEC, the parameter information is discovered for DB2 Cloning Tool, and the Customizer Workplace panel is displayed.

Results

The discovered parameter values for DB2 Cloning Tool replace any existing values.

What to do next

The next step depends on your environment:

- If DB2 entries were not discovered, or if you need to customize DB2 Cloning Tool on new DB2 entries, create and associate the entries.
- If DB2 entries were discovered and you want to customize DB2 Cloning Tool on only these entries, define the parameters.

Related tasks:

“Creating and associating DB2 entries” on page 58
 You can create new DB2 entries and associate them with DB2 Cloning Tool.

“Defining parameters” on page 59

To customize DB2 Cloning Tool, you must define DB2 Cloning Tool parameters and DB2 parameters, if your customization requires DB2 entries.

Creating and associating DB2 entries

You can create new DB2 entries and associate them with DB2 Cloning Tool.

About this task

The list of associated DB2 entries is on the Customizer Workplace panel.

Procedure

1. Issue the ASSOCIATE command on the Customizer Workplace panel. The Associate DB2 Entry for Product panel is displayed, as shown in the following figure:

```
CCQPDA          Associate DB2 Entry for Product          16:17:02
Command ==>                               Scroll ==> CSR

Select any of the following DB2 entries to add them to the Customizer
Workplace panel. You use the Customizer Workplace panel to choose the DB2
subsystems, data sharing members, and group attach names on which to
customize the product.

Commands: CREATE - Create a new DB2 entry

Product to Customize
Product metadata library : CKZ.V310.SCKZDENU          > LPAR . . : RS22
Product name . . . . . : DB2 Cloning Tool for z/OS
Product version . . . . : 3.1.0

Line commands: A - Associate C - Copy

Cmd SSID GrpAttch
----- End of DB2 entries -----
```

Figure 4. The Associate DB2 Entry for Product panel

2. Create DB2 entries. If you need to associate DB2 entries that are already in the master list, skip this step and go to step 3.
 - a. Issue the CREATE command. The Create DB2 Entries panel is displayed, as shown in the following figure:

```
CCQPDCR          Create a DB2 Entry
Command ==>

Specify a DB2 subsystem ID, a DB2 group attach name, or both for the
new DB2 entry. Press Enter to continue or End to cancel.

New DB2 Entry Information
DB2 subsystem ID . . . . .
DB2 group attach name . .
```

Figure 5. The Create a DB2 Entry panel

- b. In the appropriate columns, specify a DB2 subsystem ID, DB2 group attach name, or DB2 data sharing member name for the DB2 entry that you want to create, and press Enter. Valid values are 1 - 4 characters. You can use symbolic characters. You cannot use blanks.

Tips:

- To insert multiple DB2 entries, specify the *Inn* line command, where *nn* is the number of DB2 entries to be inserted.
- You will define specific parameters for these new DB2 entries on the DB2 Parameters panel. This panel is displayed after you select these new DB2 entries and issue the line command to generate the jobs, after you issue the primary command to generate the jobs for all associated DB2 entries, or when you manually edit the DB2 parameters.

The Associate DB2 Entry for Product panel is displayed, and the new DB2 entry is displayed in the master list, as shown in the following figure:

```

CCQPDAD          Associate DB2 Entry for Product          Row 1 to 1 of 1
Command ==>>>                                     Scroll ==>> CSR

Select any of the following DB2 entries to add them to the Customizer
Workplace panel. You use the Customizer Workplace panel to choose the DB2
subsystems, data sharing members, and group attach names on which to
customize the product.

Commands: CREATE - Create a new DB2 entry

Product to Customize
Product metadata library : CKZ.V310.SCKZDENU          > LPAR . . . : RS22
Product name . . . . . : DB2 Cloning Tool for z/OS
Product version . . . . . : 3.1.0

Line commands: A - Associate  C - Copy

Cmd SSID GrpAttch
DB02
----- End of DB2 entries -----

```

Figure 6. The Associate DB2 Entry for Product panel with a new DB2 entry in the master list

- c. Repeat steps b and c for each DB2 entry that you want to create.
 - d. When you have created all the DB2 entries, associate them with DB2 Cloning Tool, or press End to display the Customizer Workplace panel.
3. Associate DB2 entries.
 - a. Specify A against one or more DB2 entries in the master list, and press Enter to associate them with DB2 Cloning Tool.

Results

The Customizer Workplace panel is displayed with the associated DB2 entries displayed in the associated list.

What to do next

Define the parameters.

Related concepts:

“Tools Customizer terminology” on page 529

Tools Customizer uses several unique terms that you should be familiar with before you begin to use Tools Customizer.

Defining parameters

To customize DB2 Cloning Tool, you must define DB2 Cloning Tool parameters and DB2 parameters, if your customization requires DB2 entries.

About this task

You must define the DB2 Cloning Tool parameters first for the following reasons:

- If you ran the DB2 Cloning Tool Discover EXEC, you must review the values that were discovered.
- If you select optional tasks and steps on the Product Parameters panel that affect the DB2 entry on which you will customize DB2 Cloning Tool, additional parameters might be displayed on the DB2 Parameters panel.
- If other steps must be completed in a specific sequence, customization notes on the Product Parameters panel will display the correct sequence.

Defining DB2 Cloning Tool parameters

DB2 Cloning Tool parameters are specific to DB2 Cloning Tool.

About this task

If you ran the DB2 Cloning Tool Discover EXEC, you must review the parameters that were discovered.

Procedure

1. Specify E next to the **Product parameters** field on the Customizer Workplace panel, and press Enter. The Product Parameters panel is displayed, as shown in the following figure. If other steps must be completed in a specific sequence before you define the DB2 Cloning Tool parameters, a note labeled **Important** will display the correct sequence on this panel.

```
CCQPPRD                               Product Parameters                               16:21:11
Command ==>                           Scroll ==> CSR

Complete the following tasks to customize the products. The required tasks and
steps are preselected. Ensure that all parameters are specified for each
selected step within a task. Press End to save and exit.

Commands: SAVE - Save parameter values
Line Commands: / - Select

Product to Customize
  Product metadata library : CKZ.V310.SCKZDENU           > LPAR. . . : RS22
  Product name . . . . . : DB2 Cloning Tool for z/OS    > Version . . : 3.1.0

Product customization library : TWUSR.TCZ.$RS22$.CKZ310
                                                    More:      +

Required parameters
  HLQ of DB2 Cloning Tool installation . . . CKZ.CKZ310

/ Run INIMERGE

/ Run INIMERGE
  Location of the CKZINI currently in use PREVIOUS-RELEASE      List...
  HLQ of previous release DB2 Cloning Tool installation
  CKZ.CKZ220

/ Subsystem Cloning

/ Run Subsystem Cloning plan bind
  DB2 plan name . . . . . CKZPLAN
```

Figure 7. The Product Parameters panel

2. Select any required tasks and steps, and specify values for any parameters. After you select a task or step with a slash (/), put the cursor in the selected

field and press Enter. If tasks, steps, and parameters are required, they are preselected with a slash (/). Otherwise, they are not preselected.

All of the required parameters have default values, which you can either accept or change.

Tips:

- In the command line, specify the KEYS command, and map EXPAND to one of the function keys.
 - For a detailed description of all input fields, put the cursor in the field, and press F1 or the key that is mapped to Help.
 - The following elements apply to specific fields:
 - **Add...** is displayed when parameters can have multiple values but currently have only one value. To specify multiple values in these fields, place the cursor on **Add...**, and press Enter. Use the displayed panel to add or delete additional values.
 - **List...** is displayed when the complete list of valid values for the fields is too long to be displayed on the panel. To see the complete list of values, place the cursor on **List...**, and press F1 or the key that is mapped to Help.
 - **More...** is displayed when input fields contains multiple values. To see all of the values in the field, place the cursor on **More...**, and press Enter.
3. Optional: Select other tasks and steps with a slash (/) and press Enter to activate the input fields. Either accept or change the default values that are displayed.
 4. Press End to save your changes and exit, or issue the SAVE command to save your changes and stay on the Product Parameters panel.

Results

The Customizer Workplace panel is displayed, and the status of the product parameters is Ready to Customize.

What to do next

If the status of other parameters on the Customizer Workplace panel is Incomplete or Discovered, edit these parameters.

Related tasks:

“Defining DB2 parameters”

DB2 parameters are parameters for a DB2 entry.

Defining DB2 parameters

DB2 parameters are parameters for a DB2 entry.

About this task

If you did not run the DB2 Cloning Tool Discover EXEC, you must create and associate one or more DB2 entries before you can define the DB2 parameters. For more information, see “Creating and associating DB2 entries” on page 58.

Procedure

1. Specify E next to one or more DB2 entries in the associated list, which is in the Associated DB2 Entries and Parameter Status section on the Customizer Workplace panel, and press Enter. The DB2 Parameters panel is displayed, as

shown in the following figure:

```
CCQPDB2                DB2 Parameters                16:23:05
Command ==>>>          Scroll ==>> CSR
Enter values for all of the DB2 parameters. Press End to save and exit.

Commands: SAVE - Save parameter values

Product to Customize
Product metadata library : CKZ.V310.SCKZDENU          > LPAR. . . : RS22
Product name . . . . . : DB2 Cloning Tool for z/OS > Version . . : 3.1.0

DB2 subsystem ID . . . . . : DA1A
Group attach name . . . . . :

General DB2 Information
Mode . . . . . NFM (CM,CM8,CM9,NFM)
Level Number . . . . . 101 (810,910,101)

DB2 Libraries
Load Library . . . . . DSN.SDSNLOAD          > Add...
Exit Library . . . . . DSN.SDSNEXIT         > Add...

DB2 Cloning Tool parameters
DB2 package name . . . . . CKZPACK
```

Figure 8. The DB2 Parameters panel

2. Specify values for all parameters that are displayed.

Tips:

- In the command line, specify the KEYS command, and map EXPAND to one of the function keys.
- For a detailed description of all input fields, put the cursor in the field, and press F1 or the key that is mapped to Help.
- The following elements apply to specific fields:
 - **Add...** is displayed when parameters can have multiple values but currently have only one value. To specify multiple values in these fields, place the cursor on **Add...**, and press Enter. Use the displayed panel to add or delete additional values.
 - **List...** is displayed when the complete list of valid values for the fields is too long to be displayed on the panel. To see the complete list of values, place the cursor on **List...**, and press F1 or the key that is mapped to Help.
 - **More...** is displayed when input fields contains multiple values. To see all of the values in the field, place the cursor on **More...**, and press Enter.

Many parameters have default values, which you can either accept or change.

3. Press End to save your changes and exit, or issue the SAVE command to save your changes and stay on the same panel.

Results

The status of the DB2 entries that you selected on the Customizer Workplace panel is Ready to Customize.

What to do next

If the status of other parameters on the Customizer Workplace panel is Incomplete or Discovered, edit these parameters.

Related tasks:

“Defining DB2 Cloning Tool parameters” on page 60
DB2 Cloning Tool parameters are specific to DB2 Cloning Tool.

Generating customization jobs

To generate customization jobs for DB2 Cloning Tool and any associated DB2 entries, issue the `GENERATEALL` command, or select one or more DB2 entries on which to customize DB2 Cloning Tool.

Procedure

Generate the customization jobs by using one of the following methods.

- If you want to generate customization jobs at the product level and for any associated DB2 entries, issue the `GENERATEALL` command, and press Enter.
- If you want to generate customization jobs for specific DB2 entries, select the DB2 entries by specifying the `G` line command against them, and press Enter. The available DB2 entries are in the associated list in the Associated DB2 Entries and Parameter Status section.

Important: Regenerating customization jobs will replace any existing jobs, including jobs that you might have manually modified after they were generated.

Results

If the status is Incomplete or Discovered for DB2 Cloning Tool parameters or DB2 parameters, Tools Customizer automatically starts an editing session for the types of parameters that are required. The session continues until the panel for each type of required parameter has been displayed.

What to do next

If an automatic editing session is started, accept the displayed parameter values or define values for the required types of parameters, select optional parameters, tasks, or steps for your environment, and save the parameter values. Otherwise, the customization jobs are generated, and you can submit them.

Tip: If the customization jobs are generated, but you are not ready to submit them, you can see them later by issuing the `JOBLIST` command on the Customizer Workplace panel. The `JOBLIST` command displays the Finish Product Customization panel, which you can use to submit the jobs.

Submitting customization jobs

Submit the customization jobs to customize DB2 Cloning Tool.

Before you begin

Ensure that the correct jobs are generated.

About this task

The following figure shows part of the Finish Product Customization panel. The table on this panel shows the customization jobs that are generated by Tools

Customizer. They are grouped by job sequence number.

```

CCQPCST          Finish Product Customization          Row 1 to 11 of 11
Command ==>>>                                     Scroll ==>>> CSR

Submit the members in the order in which they apply to all DB2 entries. To
submit the job, browse the member and issue the TSO SUBMIT command, or browse
the customized library and submit the jobs from there.

Product to Customize
  Product metadata library : CKZ.V310.SCKZDENU          > LPAR . . : RS22
  Product name . . . . . : DB2 Cloning Tool for z/OS   > Version . : 3.1.0

Line Commands: E - Edit  B - Browse

  Product customization library . : TWUSR.TCZ.$RS22$.CKZ310          >

Cmd Member  SSID GrpAttch Template Date      Description
-----
A0IMRG  DA1A  --      CKZIMRG  2011/07/19  Run INIMERGE to merge a previou
A1BNDPAA DA1A  --      CKZBNDPS 2011/07/19  Subsystem Cloning plan bind
A2BNDCAA DA1A  --      CKZBNDSS 2011/07/19  Subsystem Cloning package bind
A4DEFSAA DA1A  --      CKZDEFSP 2011/07/19  Define Subsystem Cloning stored
A5BNDCAA DA1A  --      CKZBNDSP 2011/07/19  Bind the Subsystem Cloning stor
A6BNDPAA DA1A  --      CKZBNDPT 2011/07/19  Table Space Cloning plan bind
A7BNDTAA DA1A  --      CKZBNDTS 2011/07/19  Table Space Cloning package bin
A8BNDTAA DA1A  --      CKZBNDTD 2011/07/19  Table Space Cloning package bin
A9TS1   DA1A  --      CKZTS1   2011/07/19  Allocate Table Space Cloning ru
B0TS2   DA1A  --      CKZTS2   2011/07/19  Allocate Table Space Cloning da
B1TS3   DA1A  --      CKZTS3   2011/07/19  Create XML create DDL member
----- End of customized jobs -----

```

Figure 9. The Finish Product Customization panel

The member-naming conventions depend on whether the customization jobs are for DB2 entries, and LPAR, or the product.

Customization jobs for DB2 entries

The members use the following naming convention:

`<job_sequence_number><job_ID><DB2_entry_ID>`

where

job_sequence_number

Two alphanumeric characters, A0 - Z9, that Tools Customizer assigns to a customization job. The number for the first template in the sequence is A0, the number for the second template is A1, and so on.

job_ID

Characters 4 - 7 of the template name, if the template name contains five or more characters. Otherwise, only character 4 is used. DB2 Cloning Tool assigns the template name.

DB2_entry_ID

Two alphanumeric characters, AA - 99, that Tools Customizer assigns to a DB2 entry.

For example, the XYZBNDD*DB2_entry_ID_1* and XYZBNDD*DB2_entry_ID_2* jobs are generated from the XYZBNDGR template, and the XYZ4*DB2_entry_ID_1* and XYZ4*DB2_entry_ID_2* jobs are generated from the XYZ4 template. If the jobs are generated on two DB2 entries, the following member names are listed sequentially: A0BNDGAA, A0BNDGAB, A14AA, A14AB.

Customization jobs for the product

The members use the following naming convention:

<job_sequence_number><job_ID>

where

job_sequence_number

Two alphanumeric characters, A0 - Z9, that Tools Customizer assigns to a customization job. The number for the first template in the sequence is A0, the number for the second template is A1, and so on.

job_ID Characters 4 - 8 of the template name, if the template name contains five or more characters. Otherwise, only character 4 is used. For example, for the XYZMAKE template, the job ID is MAKE. For the XYZM template, the job ID is M. DB2 Cloning Tool assigns the template name, and it is displayed in the Template column.

For example, the XYZBNDGR job is generated from the XYZBNDGR template, and the XYZ4 job is generated from the XYZ4 template. The following member names are listed sequentially: A0BNDGR, A14.

Procedure

1. Submit the generated customization jobs by following the process that you use in your environment or by using the following method:
 - a. Specify B against a customization job or the product customization library, and press Enter. An ISPF browsing session is started.
 - b. Browse the customization job or each member in the library to ensure that the information is correct.
 - c. Run the TSO SUBMIT command.
2. Press End.

Results

DB2 Cloning Tool is customized, and the Customizer Workplace panel is displayed. The status is Customized for the DB2 entries on which DB2 Cloning Tool was customized.

What to do next

You can generate more customization jobs for other DB2 entries, view a list of customization jobs that you previously generated, or recustomize DB2 Cloning Tool.

Browsing parameters

You can browse the product parameters and the DB2 parameters in read-only mode.

Procedure

1. On the Customizer Workplace panel, specify B next to the **Product parameters** field or the DB2 entry that you want to browse, and press Enter. The panel that corresponds to your specification is displayed.
2. Press End to exit.

Copying DB2 entries

You can copy associated and not associated DB2 entries to other DB2 entries or to new DB2 entries.

About this task

Go to the step that applies to your environment:

- To copy an associated DB2 entry to another associated DB2 entry or to an entry that is not associated, go to step 1.
- To copy an associated DB2 entry to a new entry, go to step 2.
- To copy a DB2 entry that is not associated to a new entry, go to step 3.

Procedure

1. To copy an associated DB2 entry to another associated DB2 entry or to an entry that is not associated, complete the following steps:
 - a. Specify C against a DB2 entry in the associated list of DB2 entries on the Customizer Workplace panel, and press Enter. The Copy Associated DB2 Entry panel is displayed.
 - b. Select one or more DB2 entries to which information will be copied by specifying the / line command, and press Enter. The Associated column indicates whether the DB2 entry is associated.

Tip: To copy information into all of the DB2 Entries in the list, issue the SELECTALL primary command, and press Enter. The Copy DB2 Parameter Values panel is displayed.

- c. Specify an option for copying common and product-specific DB2 parameter values. Common DB2 parameter values apply to all DB2 entries for all products that you have customized by using Tools Customizer. Product-specific DB2 parameter values apply only to the product that you are currently customizing.
 - To copy the common DB2 parameter values and the product-specific DB2 parameter values, specify option 1, and press Enter.
 - To copy only the product-specified DB2 parameter values, specify option 2, and press Enter.

In some cases, the DB2 parameter values might contain the DB2 subsystem ID as an isolated qualifier in data set names. For example, in the DB01.DB01TEST.DB01.SANLLOAD, data set name, the DB01 subsystem ID is isolated in the first and third qualifiers but is not isolated in the second qualifier. When the DB2 subsystem ID is an isolated qualifier in data set names, the Change DB2 Subsystem ID in DB2 Parameter Values panel is displayed. Otherwise, the Customizer Workplace panel is displayed.

- d. If the Change DB2 Subsystem ID in DB2 Parameter Values panel is displayed, specify an option for changing the subsystem IDs. Otherwise, skip this step.
 - To change the subsystem ID in isolated qualifiers in data set names, specify option 1, and press Enter.
 - To use the same subsystem ID in all values, specify option 2, and press Enter.

The Customizer Workplace panel is displayed with the copied associated entry in the list.

2. To copy an associated DB2 entry to a new entry, complete the following steps:

- a. Specify C against a DB2 entry in the associated list of DB2 entries on the Customizer Workplace panel, and press Enter. The Copy Associated DB2 Entry panel is displayed.
 - b. Issue the CREATE command. The Create DB2 Entries panel is displayed.
 - c. Specify the SSID, the group attach name, or both in the appropriate columns for each new DB2 entry, and press Enter.

Tip: To add rows for additional entries, specify the *Inn* line command, where *nn* is the number of entries to be created, and press Enter. The Copy Associated DB2 Entry panel is displayed with the new entries in the list. The new entries are preselected.
 - d. Press Enter to complete the copy process. The Customizer Workplace panel is displayed with the copied entries in the list.
3. To copy a DB2 entry that is not associated to a new entry, complete the following steps:
- a. Issue the ASSOCIATE command on the Customizer Workplace panel. The Associate DB2 Entry for Product panel is displayed.
 - b. Select one or more DB2 entries by specifying the / line command, and press Enter. The Copy a DB2 Entry panel is displayed.
 - c. Specify the SSID, the group attach name, or both in the appropriate columns for the new DB2 entry, and press Enter. The Associate DB2 Entry for product panel is displayed with the copied entry in the list.
 - d. If you want to associate the copied entry, specify A against it, and press Enter. The Customizer Workplace panel is displayed with the copied entries in the list.

What to do next

Edit any of the parameters or generate the jobs.

Related concepts:

“Tools Customizer terminology” on page 529

Tools Customizer uses several unique terms that you should be familiar with before you begin to use Tools Customizer.

Removing DB2 entries

You can remove DB2 entries from the associated list.

About this task

When you remove DB2 entries from the associated list, any customization jobs for the entries are removed from the list of jobs on the Finish Product Customization panel, and they are deleted.

Procedure

On the Customizer Workplace panel, specify R next to one or more DB2 entries that you want to remove, and press Enter. The selected DB2 entries are removed from the associated list and added to the master list on the Associate DB2 Entry for Product panel, and the customization jobs are deleted.

Related concepts:

“Tools Customizer terminology” on page 529

Tools Customizer uses several unique terms that you should be familiar with

before you begin to use Tools Customizer.

Deleting DB2 entries

You can delete DB2 entries from the master list.

About this task

When you delete DB2 entries from the master list, any associations and all customization jobs for products that are customized on the entries will be deleted.

Procedure

1. On the Customizer Workplace panel, issue the ASSOCIATE command. The Associate DB2 Entry for Product panel is displayed.
2. Specify D next to one or more DB2 entries that you want to delete, and press Enter. If the entry is associated with any products, the Delete Associated DB2 Entry panel for the first DB2 entry that you selected is displayed. Otherwise, the Delete DB2 Entry panel is displayed.
3. To delete the DB2 entries, press Enter. If the DB2 entries are associated with any products in the table on the Delete Associated DB2 Entry panel, any associations and all customization jobs for the products that are customized on it are deleted. Otherwise, only the DB2 entries are deleted. If you selected multiple DB2 entries to delete, the next DB2 entry that you selected is displayed on either the Delete Associated DB2 Entry panel or the Delete DB2 Entry panel. Otherwise, the Associate DB2 Entry for Product panel is displayed.

What to do next

If you selected multiple DB2 entries to delete, repeat step 3 until all selected entries are deleted. Then, continue the customization process.

Displaying customization jobs

You can view a list of the members that contain the customization jobs before or after you submit the jobs.

About this task

The customization jobs that you generate for one DB2 entry are also displayed when you customize DB2 Cloning Tool for another DB2 entry later.

Procedure

On the Customizer Workplace panel, issue the JOBLIST command. The Finish Product Customization panel is displayed. This panel shows the list of jobs that you have previously generated. They are grouped by job sequence number. Use this panel to browse or edit the generated jobs before you submit them.

Maintaining customization jobs

Instead of deleting customization jobs outside of Tools Customizer, you can maintain the correct jobs for DB2 Cloning Tool by completing the steps for recustomization.

About this task

You cannot delete or rename customization jobs from the customization library by starting an ISPF browse or edit session from the Finish Product Customization panel. If you try to delete customization jobs by using this method, the CCQC034S message is issued. If you try to rename customization jobs, the CCQC035S message is issued.

If you delete or rename customization jobs from the customization library by using ISPF outside of Tools Customizer, Tools Customizer will not recognize that the jobs were deleted, and the Finish Product Customization panel will still display them. If you browse or edit jobs that were deleted from the library outside of Tools Customizer, the CCQC027S message is issued.

Procedure

To maintain the correct customization jobs in the customization library, complete the steps for recustomization.

Using Tools Customizer in a multiple-LPAR environment

Currently, Tools Customizer supports only the local LPAR; however, you can propagate customizations to additional LPARs by using either of two different methods.

About this task

In a multiple-LPAR environment, Tools Customizer identifies the LPAR to which you are logged on. Tools Customizer uses this LPAR name for several different parameter settings, one of which is the data store. When you use the data store during the customization of DB2 Cloning Tool that is on a different LPAR, Tools Customizer issues message CCQD586S, which indicates that the product has already been customized based on values from the data store on the first LPAR. This message is issued to prevent the data store from becoming corrupted.

This behavior occurs in the following conditions:

- Tools Customizer is installed on a DASD device that is shared by multiple LPARs.
- After a product is customized by using Tools Customizer, the data store is copied to another LPAR.

Procedure

To customize products running against a DB2 subsystem on an LPAR where Tools Customizer is not installed, consider using one of the following methods:

Install one instance of Tools Customizer on one LPAR

If you intend to reuse the customization values for all the instances of your products on all LPARs, use this method.

1. Associate all the DB2 entries in this one instance of Tools Customizer. The LPARs on which the DB2 subsystems reside do not matter.
2. Generate the customization jobs for each DB2 entry.
3. Copy the generated customization jobs to the LPAR to run against the specific DB2 entries. Some LPAR-specific edits might be required. You can make these edits in the customized jobs that you copied. Note that

this situation is one of the few situations where you might need to make manual changes to the jobs that are customized by Tools Customizer.

Install one instance of Tools Customizer on each LPAR

If you do not want to reuse previous customization values and you want to start new customizations, use this method.

Important: This method will likely not be the preferred approach for most organizations because most organizations tend to use similar or identical customization values for each product instance on all LPARs.

Chapter 5. After customizing DB2 Cloning Tool using Tools Customizer

The following topics describe optional tasks that might be completed immediately after DB2 Cloning Tool is customized and tasks that might need to be performed if you migrate to a new version of DB2 after customizing DB2 Cloning Tool.

Reviewing the CKZINI PARMLIB member variables

The keywords in the CKZINI PARMLIB member provide flexibility and control of product execution, and security of system parameters. Make sure to review and update the generated CKZINI member to ensure all keywords are appropriate for your installation and operation requirements.

You can alter the keyword values by directly updating the CKZINI member. This member is a read-only library during execution of the product.

“CKZINI customization values” on page 575 contains descriptions and coding rules for the keywords in the CKZINI member.

Use ISPF Edit to update the CKZINI configuration values.

Some keywords may be coded with default values. You should review the following keywords in particular, but make sure to review all of the CKZINI to ensure the values are correct for your installation:

- Verify the DB2 plan name(s) in the DB2_OPTIONS section. All DB2 subsystems that DB2 Cloning Tool and DB2 Cloning Tool Table Space Cloning will access must have a plan defined. This plan name is provided to DB2 Cloning Tool and DB2 Cloning Tool Table Space Cloning using the DB2_PLAN value.
- Verify the CA-MIM/MII token values in RESOURCE_SERIALIZATION section. Installations running CA-MIM/MII with multiple systems and shared DASD need to set the following CKZINI parameter found in the :RESOURCE_SERIALIZATION section to YES to ensure that when CA-MIM/MII GDIF is inactive, the DB2 Cloning Tool data sets are protected from data corruption.
:RESOURCE_SERIALIZATION
MIM_GDIF = YES
- Verify the TCP/IP name in the TCPIP-OPTIONS section. If you are planning to use the TCP/IP option of DB2 Cloning Tool Table Space Cloning and your TCP/IP started task name is not TCPIP, update the TCPIP_STC_NAME to match your installation's started task name.

Managing CKZINI when running DB2 Cloning Tool on several systems

For those customers with DB2 Cloning Tool running on several systems, the CKZINI PARMLIB member can be managed either by:

- Using separate copies of the CKZINI, editing them independently and running INIMERGE on each individually.
- Using the same text (multi-image INI) for all systems; this requires replicating sections with sysplex and/or system names qualifications for those sections that

need token/values unique to an image. Refer to “CKZINI customization values” on page 575 for more information related to qualified section names. Use the CKZIVIEW member of the SCKZJCL data set to view the contents of the multi-image INI that will be used by any image.

Verifying successful customization

When customization is completed, you can verify that DB2 Cloning Tool and DB2 Cloning Tool Table Space Cloning have been successfully installed.

Once the customization tasks have been completed, the FINDUCATS command can be run against one volume to verify that DB2 Cloning Tool has been successfully customized. To verify that DB2 Cloning Tool Table Space Cloning has been successfully installed, select a DB2 table space and all its index spaces; copy it, and ensure it can be accessed on the target subsystem.

If you have configured the ISPF interface, verify successful customization of the interface by running the CKZCLIST CLIST.

Adding DB2 Cloning Tool to the DB2 Administration Tool Launchpad

DB2 Cloning Tool can be added to the DB2 Administration Tool for z/OS Launchpad. The DB2 Admin Launchpad allows you to launch installed IBM DB2 tools directly from a centralized panel.

Before you begin

This procedure assumes that the DB2 Administration Tool (also called DB2 Admin) has been installed at your site and the Launchpad table has already been created. For detailed information about preparing and using the DB2 Admin Launchpad, refer to the DB2 Administration Tool User's Guide, available on the Web at the following web site: <http://www.ibm.com/software/data/db2imstools/db2tools-library.html>.

About this task

This procedure is only required if you want to start DB2 Cloning Tool from the DB2 Admin Launchpad.

Procedure

1. Locate the sample REXX EXEC called CKZADBI in the SCKZJCL library.
2. Following the instructions in the sample EXEC to modify it.
3. Run the EXEC by executing the following command: EXEC '*dsname*(CKZADBI) '
where *dsname* is the data set name that contains the EXEC.

Results

DB2 Cloning Tool is added to the Launchpad and you can start DB2 Cloning Tool from the DB2 Admin Launchpad Table.

DB2 version migration considerations

Once DB2 Cloning Tool has been installed and configured, follow these steps if a DB2 subsystem being used with DB2 Cloning Tool is migrated to a later version or must fallback to a previous version.

DB2 Cloning Tool Subsystem Cloning

During configuration, the DB2 subsystem cloning plan and package should have been bound as part of the customization process. The plan and package are release-independent; therefore, once the plan and package is bound, no further action is required when a DB2 subsystem is migrated.

DB2 Cloning Tool Table Space Cloning

When migrating from one DB2 version to a later version, a rebind on the new subsystem is required for the plans and packages that were used by DB2 Cloning Tool Table Space Cloning on DB2 subsystems. Rebind the plans and packages on each subsystem on which DB2 Cloning Tool Table Space Cloning will be used. In general, the steps are:

1. Start Tools Customizer and specify to customize DB2 Cloning Tool.
2. On the Customizer workplace panel, edit the product parameters.
3. On the Product Parameters panel, select the Run Table Space Cloning plan bind task, the Run Table Space Cloning package binds task, and optionally the Run Table Space Cloning package binds for DDF location task. Save and exit.
4. Create and associate the new DB2 subsystem.
5. On the Customizer Workplace panel, edit the new subsystem.
6. On the DB2 Parameters panel, specify the mode, level, and other required DB2 subsystem parameters. Save and exit.
7. Generate the jobs for the subsystem.

Refer to “Roadmap: Recustomizing DB2 Cloning Tool” on page 53 for additional information.

Fallback instructions

If after migrating to a later DB2 version, you must fall back to the prior version of DB2:

- For DB2 Cloning Tool Subsystem Cloning, the plan and package are release-independent; therefore, once the plan and package is bound, no further action is required.
- For DB2 Cloning Tool Table Space Cloning, the plan and packages must be rebound on each subsystem on which DB2 Cloning Tool Table Space Cloning will be used. In general, the steps are:
 1. Start Tools Customizer and specify to customize DB2 Cloning Tool.
 2. On the Customizer workplace panel, edit the product parameters.
 3. On the Product Parameters panel, select the Run Table Space Cloning plan bind task, the Run Table Space Cloning package binds task, and optionally the Run Table Space Cloning package binds for DDF location task. Save and exit.
 4. If not already done, create and/or associate the prior version DB2 subsystem.
 5. On the Customizer Workplace panel, edit the prior version subsystem.
 6. On the DB2 Parameters panel, specify the mode, level, and other required DB2 subsystem parameters. Save and exit.
 7. Generate the jobs for the subsystem.

Refer to “Roadmap: Recustomizing DB2 Cloning Tool” on page 53 for additional information.

Chapter 6. Planning for copying and renaming volumes

Before attempting to actually use the DB2 Cloning Tool process, some planning and decision-making should take place. This topic discusses those things that need to be considered to clone a DB2 subsystem or a volume.

Selection of source and target volumes

The following requirements should be considered when planning the cloning environment.

Scope of source volumes to be copied

The source volumes copied must include all data sets required by the application that will access the renamed data sets on the target volumes. It is recommended that the source DB2 SDSNLOAD and SDSNEXIT libraries on source volumes should not be cloned into the target DB2 SDSNLOAD and SDSNEXIT libraries. The reasons for this recommendation are:

- There may be problems setting up APF authorization of the target SDSNLOAD and SDSNEXIT libraries if the target volumes are not SMS managed and DB2 Cloning Tool is used to do the copy. The target volumes these libraries get cloned to might be different for different runs.
- Member DSNHDECP has the default DB2 SSID in it. This module can only have one name and the default SSID should be different between the source and target DB2.

The DB2 directory, DB2 catalog, active logs, and BSDSs must be contained on the source volumes, as well as the application databases.

Desired target volume data sets

Any copies of data sets to be used by the application must reside within the list of "cloned" volumes, and must match a rename mask. DB2 Cloning Tool does not require all data sets on target volumes be renamed. The NOTRENAMED parameter is used to specify:

- The disposition of any data sets not matched to a rename mask.
- The return code if at least one data set is not renamed.

If all data sets on all copied volumes are critical, a return code of eight (8) should be requested, as an indication that the errors must be reviewed and the process rerun. This assumes that a return code of eight (8) is treated by the application as a RENAME failure.

If rename masks intentionally match only some data sets, specify a return code of zero (0) or four (4). Normally a disposition of DELETE should be specified if the volumes are SMS managed (in keeping with SMS rules), and/or if the space occupied by not-renamed data sets may be needed for subsequent allocations.

Note: By not renaming all data sets, the process is essentially a selective data set copy, with the caveat that those data sets renamed and kept occupy the same track locations as their source volume counterparts.

DB2 Cloning Tool deletes, on the target volumes, any temporary data sets created from the volume copies. DB2 Cloning Tool does *not* detect, catalog, or delete any data sets on target volumes whose source volume counterparts are uncataloged. Despite not being cataloged, the target volume data set will be renamed, if it matches a rename mask.

Note: For performance reasons, DB2 Cloning Tool does not diagnose from volume to BCS to detect anomalies, such as uncataloged data sets, especially when the anomaly can be replicated each cycle of the process. It is recommended that volumes and catalogs be diagnosed routinely to identify and repair the source of any problems.

An ICF catalog can be renamed, but it will not be usable as an ICF catalog.

Source and target volume condition

To prevent volume internal VTOC index and VVDS errors, the VTOC, VTOCIX, and VVDS should be in the same location on the target volumes as they are on the source volumes prior to the COPY taking place.

Target volume online status

DB2 Cloning Tool expects that the target volumes will be online to only one system while RENAME runs. If the target volumes are online to more than one system, VTOC index and VVDS errors may occur.

Data set/sphere integrity

By default, DB2 Cloning Tool will detect and fail the process if one of the data set integrity violations listed in the following table is encountered:

Table 14. Data set/sphere integrity violations

Integrity violation	Description
Multivolume data sets	All segments of multivolume data sets, where the source volume data set matches a rename mask, must be wholly contained within the list of source volumes. When we use the term 'multivolume data sets', multivolume data sets, striped data sets, and VSAM spheres are implied
VSAM spheres	All components and associations of VSAM spheres, where the cluster name matches a rename mask, cannot be partially on the source volumes. This means the volume list must include all components of a cluster and any alternate indexes and their components that are associated with the base.
GDGs	All active generations of a GDG base, where the base name matches a rename mask, cannot be partially on the source volumes. Exceptions are allowed for active generations that are on tape or have been migrated, but references to such target entries will fail.
Non-VSAM data set aliases	If a data set found on a volume matches a rename mask, and the data set has one or more aliases, all aliases must also match some rename mask.

Discrete volumes vs. SMS storage group specification

Specification of volumes by SMS storage groups (available only for FlashCopy, SnapShot, or TimeFinder/Clone) will reduce maintenance required to ensure that

volumes used by an application are included in the FlashCopy, SnapShot, or TimeFinder/Clone list. If storage group specification is used, exclude options are available to exclude source or target volumes from the storage group list.

For example, there is a source SMS storage group SRCSG1 and a target SMS storage group TGTSG1. SMS storage group SRCSG1 has two volumes: SRC001 and SRC002. SMS storage group TGTSG1 has two volumes: TGT001 and TGT002. The COPY is using FROM-STORAGEGROUP(SRCSG1) and TO-STORAGEGROUP(TGTSG1). COPY would use as source volumes SRC001 and SRC002, and would use TGT001 and TGT002 as target volumes.

Now volume SRCP01 is added to SRCSG1 and volume TGTP01 is added to TGTSG1. Now COPY would use as source volumes SRC001, SRC002, and SRCP01, and would use TGT001, TGT002, and TGTP01 as target volumes. By using SMS storage groups, the COPY statements do not have to be changed when volumes are added or removed from the SMS storage groups. COPY will use whatever volumes are currently defined in the specified SMS storage groups.

Migrated application data sets

If DB2 subsystems contain generation data sets, users must ensure that data sets to be copied are not allowed to migrate before the copy volumes are created. This includes any active generations of a renamed GDG.

If you COPY source volumes, RENAME data sets on the new target volumes, update the DB2 internals, and then start processing against the target volumes, your processing will fail if a job tries to access a renamed generation that was migrated at the time of COPY, and therefore not copied. Your entire process could then be compromised. DB2 Cloning Tool does NOT copy migrated data.

Requirements for source and target volume pairing

DB2 subsystem use of source volumes must be planned such that enough target volumes exist to allow all source volumes to be paired with a target volume.

FlashCopy source and target volumes:

- must be in the same subsystem
- must have the same track format (e.g., 3380 vs. 3390)
- the target volume size must be equal to or greater than the source volume

For more information on FlashCopy source and target pairing rules, contact IBM.

SnapShot source and target volumes:

- must be in the same RVA partition
- must have the same track format (e.g., 3380 vs. 3390)
- the target volume size must be equal to or greater than the source volume

For more information on SnapShot source and target pairing rules, contact StorageTek.

EMC TimeFinder/Clone source and target volumes:

- must be in the same subsystem
- must have the same track format (e.g., 3380 vs. 3390)
- the target size must be equal to or greater than the source volume

For more information on TimeFinder/Clone source and target pairing rules, contact EMC.

Volume relationship conflicts

Care should be taken to ensure that when DB2 Cloning Tool initiates FlashCopy, SnapShot, or TimeFinder/Clone, no other relationships exist that would cause DB2 Cloning Tool to reject a volume, or DSS or EMCSNAP to fail.

If other relationships are desired after the DB2 Cloning Tool process is complete, a COPYCHECK command is provided to wait for all volume copies to complete.

Ensure RACF and ACF authorities

Before cloning, you should ensure that all proper RACF and ACF permissions and privileges are in place. Refer to “Verify that your environment meets security requirements” on page 19 for information about required permissions and privileges.

Data set renaming considerations

Existing naming conventions will dictate the required rename masks to ensure that all required data sets are renamed, and that renaming does not result in two or more data sets renamed to the same target name. Because DB2 Cloning Tool renaming allows introduction of additional qualifiers for the logs and BSDSs, users may wish to consider a data set name length restriction of something less than 44 bytes (35 for GDG base names).

If the RENAME masks cause the data set names to be longer than the source volume data set names, and/or the target ICF catalog name is longer than the source ICF catalog name, the VVDS on the source volume must be large enough to support the expanded target names. DB2 Cloning Tool does not add extents to target VVDS data sets.

Because of DB2 naming standard requirements, only the DB2 logs and BSDSs can be renamed beyond the *hlq*. All other DB2 files can only be renamed at the *hlq*:

- DB2 directory – DB2 expects a specific naming standard:
*hlq.DSNDBx.DSNDB01.**
- DB2 catalog – DB2 expects a specific naming standard:
*hlq.DSNDBx.DSNDB06.**
- DB2 databases – DB2 expects a specific naming standard:
vcat.DSNDBx.dbname.psname.y0001.Annn

Target ICF catalog considerations

DB2 Cloning Tool will catalog target volume data sets to either a populated or an empty ICF catalog. If a target catalog entry already exists, the RECATALOG option of the RENAME command is required. However, the BCSCLEAN command can be used to ensure this doesn't happen.

Cataloging to a populated ICF catalog may extend the execution time. When RECATALOG is specified, extra care should be taken, both initially and when maintaining the rename masks, to ensure that a target catalog entry is not replaced as the result of an incorrect target rename mask. If there are no compelling reasons

for the target catalog to hold data sets other than those resulting from a DB2 Cloning Tool process, a discrete catalog for renamed data sets is advised.

As noted in the description of the RECATALOG keyword, using the BCSCLEAN command allows target catalogs to be used by applications other than the one(s) involved with the copy process, without using the RECATALOG option. For more information, see the “BCSCLEAN” on page 345 topic.

The DB2 Cloning Tool journal data set should not be cataloged in the target ICF catalog. This can cause the DB2 Cloning Tool RENAME command to deadlock with itself over access to the target ICF catalog.

Location of the source and target ICF catalogs

Source ICF catalog information for source volume data sets being cloned is required at the point-in-time by DB2 Cloning Tool. This requirement enables the rename of the target volume data sets. Some information like VSAM sphere and GDG Base information is in the ICF catalogs only, and not on the DASD being cloned and therefore, needs to be captured at the point-in-time. Two options are available: either the source ICF catalogs can reside on source volumes being cloned, and be read from the target volumes; or the source ICF catalogs can be on non-source volumes and be read from the live source ICF catalogs at the time the replication or split occurs.

Target ICF catalogs used to catalog the renamed data sets cannot reside on a target volume during the timeframe from the volume copy through the completion of the RENAME step. If desired, you can move the target ICF catalog(s) from the target volume prior to the volume copy, and move the target ICF catalog(s) back to the target volume after the RENAME has completed.

If the target ICF catalog was on the source volume when it was cloned, it can be renamed as part of the cloning process just like any other data set, but it will not be usable as a catalog.

This example uses a cloning of one source volume to one target volume that has one source ICF catalog and one target ICF catalog, where:

- SRC001 is the source volume
- TGT001 is the target volume
- VOL001 is a volume that is not being cloned
- SRC.CATALOG is the source ICF catalog
- TGT.CATALOG is the target ICF catalog

SRC.CATALOG can be on volumes SRC001 or VOL001, but cannot be on volume TGT001. If SRC.CATALOG is on SRC001, its copy on TGT001 can be renamed, but cannot be used.

TGT.CATALOG can be on volumes SRC001 or VOL001 but cannot be on volume TGT001 from volume copy through RENAME. If it is desired to have TGT.CATALOG be on TGT001, then it must be moved to some other volume before the volume copy, and then can be moved back to TGT001 after RENAME has completed. If it is on SRC001, its copy on TGT001 can be renamed, but cannot be used.

Target data set ICF catalog aliases

Users are responsible for creating ICF catalog aliases, if needed for new target data set names.

This must be coordinated with the rename masks used, and it must be determined whether the MLA setting is different for the image(s) from where source volume data sets are accessed, compared to the MLA setting for the image(s) from where target volume data sets are to be accessed.

Note: To determine the target ICF catalogs you may wish to create, as well as the aliases needed for target volume data sets, see the FINDUCATS command. FINDUCATS identifies catalogs involved with source volume data sets and the aliases employed for source volume data sets.

Return code choices

A number of DB2 Cloning Tool options allow the user a choice of return codes if at least one circumstance is encountered for the keyword involved.

In other words, the user may choose the seriousness of a situation. Scenarios discussed in this document assume that the conditional execution of subsequent steps adheres to the convention that return code 0 means successful, 4 means warning, and 8 means an error.

The following is an example of choosing to treat a RENAME not-renamed situation as a critical error (only partial JCL and control statements included):

```
//RENAME EXEC PGM=CKZ00010
//CKZIN DD *
        RENAME -
        RENAME-MASKS(PROD.** TEST.** ) -
        NOTRENAMED(DELETE,RC(8)) -
        JOURNAL-DDN(JOURNAL)
//IFBAD IF (RENAME.RC >= 8) THEN
//TGTBAD EXEC PGM=program-to-signal-rename-failed
// ELSE
//TARGETOK EXEC PGM=application-program-that-uses-target-volumes
//IFBAD ENDIF
```

Cloning a DB2 subsystem

To clone a DB2 subsystem, the target subsystem must be set up.

Important: If utilities might be running or registered in SYSUTILX on the source DB2 subsystem when it is cloned, SYSUTILX and its indexes should be cleaned out when DB2 conditioning is complete. You can use the DB2UTILXCLEAN command for this purpose. If SYSUTILX is not cleaned out, the source DB2 subsystem might become corrupted when the target DB2 subsystem issues a **-TERM UTIL** command to clean up the entries it finds in SYSUTILX.

The following requirements also must be met:

1. The DB2 directory, DB2 catalog, active logs, and BSDSs must be contained on the source volumes that are being cloned, as well as the application databases.
2. The source DB2 SDSNLOAD and SDSNEXIT libraries on source volumes should not be cloned into the target DB2 SDSNLOAD and SDSNEXIT libraries. The reasons for this recommendation are:

- There might be problems setting up APF authorization of the target SDSNLOAD and SDSNEXIT libraries if the target volumes are not SMS managed and DB2 Cloning Tool COPY is used to do the copy. The target volumes these libraries get cloned to might be different for different runs.
 - Member DSNHDECP has the default DB2 SSID in it. This module can have only one name and the default SSID should be different between the source and target DB2.
3. Because of DB2 naming standard requirements, only the DB2 logs and BSDSs can be renamed beyond the high-level qualifier. All other DB2 files that are shown in the list that follows can be renamed only at the high-level qualifier:
 - DB2 directory – DB2 expects a specific naming standard:
*hlq.DSNDBx.DSNDB01.**
 - DB2 catalog – DB2 expects a specific naming standard:
*hlq.DSNDBx.DSNDB06.**
 - DB2 databases – DB2 expects a specific naming standard:
vcat.DSNDBx.dbname.psname.y0001.Annn

The target DB2 SDSNEXIT can be set up before the DB2 Cloning Tool processing is started. It should be an APF authorized library.

It is recommended that the target DB2 system be run with the same DB2 release and the same or similar maintenance level as the source DB2 system. The reason for this recommendation is that a different release or maintenance level of DB2 might have dependencies on the DB2 catalog, directory, or BSDS that will not exist as part of the cloning. After the cloning is complete, the target DB2 could then be migrated to a higher release or maintenance level of DB2.

The following steps can be set up before the DB2 Cloning Tool processing is started. Normally, these will be a "one-time" setup and do not have to be repeated.

1. Set up normal zparm, DSNZPARx, for the target DB2 subsystem.
Starting with a copy of the source system's zparms would be desirable. Check the macro keywords for items that might need to be changed to reflect the target subsystem. For example, CATALOG=, IRLMPRC=, and IRLMSID= would need to be changed for the target subsystem. Assemble and link-edit DSNZPARx to the target DB2 SDSNEXIT LOAD library.
2. Set up DSNHDECP for the target subsystem. The target DSNHDECP should be the same as used by the source system.
3. Assemble and link-edit DSNHDECP to the target DB2 SDSNEXIT LOAD library.
4. Set up the required started tasks JCL for your target subsystem: DSN?MSTR, DSN?DBM1, DSN?DIST, and so on.
5. Set up the target subsystem DB2 and target subsystem IRLM subsystem names in SYS1.PARMLIB, IEFSSnn.
You may use the SETSSI commands to add the target subsystem names to avoid an IPL, but ensure that the PARMLIB member is updated.
6. Set up special zparm, DSNZSPEC, for the target DB2 subsystem.
This special zparm will allow the target's DB2 catalog to be updated and defers the backout of in-flight transactions on the target subsystem. This zparm should only be used for the time needed to update the target's VCATNAMEs and, optionally, the target's DB2 storage group names.
 - a. Allocate a special macro library for DSNZSPEC. It will be a small PDS with only one member, DSN6SPRC.

b. Copy member DSN6SPRC from the distributed SDSNMACS library to the special macro library.

c. Change special macro library member, DSN6SPRC, as follows:

Change this:

```
&SPRMCTU SETC '0'          YES=CATALOG CAN BE UPDATED
```

To this:

```
&SPRMCTU SETC '1'          YES=CATALOG CAN BE UPDATED
```

d. Save the modified special macro library member. For example, the modified special DSN6SPRC macro might look something like (in part):

```
...
&SPRMNAP SETC '0'          BIT ON - SKIP ADJ. PREFETCH @KYF1570
&SPRMSHP SETC '0'          BIT ON - SIMULATE 2G HIPERSPACE
&SPRMCTU SETC '1'          YES=CATALOG CAN BE UPDATED
&SPRMXPL SETC '0'          YES=GEN ALL EXPLAIN TABLES
&SPRMNHJ SETC '0'          YES=TURN OFF HYBRID JOIN
...
```

e. Create DSNZSPEC.

This should be copied from the normal zparm, DSNZPARx, created for the target DB2 subsystem.

- Change DSNZSPEC macro DSN6SPRM from RESTART, ALL to DEFER, ALL.
- Change DSNZSPEC macro DSN6SPRM keyword SYSADM or SYSADM2 to specify the userid that will be running the SQL statements, described in Step 7, on the target DB2.
- Change the JCL for DSNZSPEC so that the special macro library is the first library in the assembly step //SYSLIB DD concatenation.
- Change all occurrences of DSNZPARx to DSNZSPEC, except on the link-edit card INCLUDE ADSNLOAD(DSNZPARM). Assemble and link-edit DSNZSPEC to the target DB2 SDSNEXIT LOAD library.

For example, the DSNZSPEC might look something like (in part):

```
...
DSN6ENV    MVS=XA
DSN6SPRM   DEFER,          X
           ALL,            X
...
           SYSADM=CKZUSER,  X
           SYSADM2=DB2ADM,  X
...
//SYSLMOD  DD  DISP=SHR,
//          DSN=target.SDSNEXIT
...
ENTRY     DSNZMSTR
NAME      DSNZSPEC(R)
...
```

7. If not already done, bind the DB2 Cloning Tool plan and package on the source subsystem.

To update the DB2 catalog on the target DB2 subsystem, the DB2 Cloning Tool DB2SQL command has a plan and package that need to be installed. If the binds for the plan and package have already been completed on the source DB2 subsystem, they will be available for use on the target DB2 subsystem when needed without having to explicitly bind them on the target DB2 subsystem after the cloning. If you choose *not* to bind the plan and package on the source subsystem, you will need to bind them on the target subsystem after cloning.

Chapter 7. Setting up to copy by volume with FlashCopy, SnapShot, or TimeFinder/Clone

Follow this procedure if volume copies are to be created with volume FlashCopy or SnapShot via DFSMSdss, or TimeFinder/Clone.

For volume copies created in other ways, see Chapter 8, “Setting up to copy by volume with an onsite mirror tool,” on page 87.

The steps to accomplish the DB2 Cloning Tool function are placed within the application's job stream, typically in multiple locations, in order for the function steps to run at the appropriate times, relative to the application's activities. The DB2 Cloning Tool steps are invoked via commands to a common program.

If multiple commands are supplied within the same step, any command that results in a return code equal to or greater than eight will terminate the step. For instance, if COPY and RENAME are in the same step, and COPY terminates with a return code 8, because insufficient target volumes were found, RENAME will be bypassed.

Several commands specific to cloning a DB2 subsystem are not mentioned here. See Chapter 9, “Cloning DB2 subsystems,” on page 95 for more information.

The available commands are:

FINDUCATS

Locates ICF user catalogs pointing to source volume data sets.

COPY Initiates copies and captures ICF catalog data.

RENAME

Renames and catalogs target volume data sets.

BCSCLEAN

Deletes catalog entries from previous executions.

DB2 Cloning Tool process for volume FlashCopy, SnapShot, or TimeFinder/Clone

The table that follows summarizes the steps required to create volume copies with volume FlashCopy, SnapShot, or TimeFinder/Clone.

Table 15. DB2 Cloning Tool process for volume FlashCopy, SnapShot, or TimeFinder/Clone

Step	Description
1	FINDUCATS step. Optional step to run one time or occasionally, to find ICF user catalogs pointing to source volume data sets.
2	Shut down or suspend the source DB2 subsystem. See Chapter 9, “Cloning DB2 subsystems,” on page 95 for additional instructions. These instructions are a prerequisite to Step 3. This step is not necessary if using consistent FlashCopy, SnapShot, or TimeFinder/Clone.
3	COPY step.

Table 15. DB2 Cloning Tool process for volume FlashCopy, SnapShot, or TimeFinder/Clone (continued)

Step	Description
4	Start up or resume the source DB2 subsystem. See Chapter 9, "Cloning DB2 subsystems," on page 95 for additional instructions. These instructions are a prerequisite to Step 5. This step is not necessary if using consistent FlashCopy, SnapShot, or TimeFinder/Clone.
5	RENAME step. This step renames and catalogs target volume data sets.
6	DB2 cloning procedures. See Chapter 9, "Cloning DB2 subsystems," on page 95 for additional instructions.
7	BCSCLEAN step. Optional step runs after target volume usage is terminated and before the next COPY.

Steps for volume FlashCopy, SnapShot, or TimeFinder/Clone

These are the steps to create volume copies with volume FlashCopy, SnapShot, or TimeFinder/Clone.

Procedure

1. **FINDUCATS Step (optional).** FINDUCATS identifies which ICF User catalogs point at data sets on the source volumes to be copied.
The COPY step requires pairs of source/target user ICF catalogs to be specified. FINDUCATS does not negate this need. It is intended to be run prior to initial setup, and possibly on an occasional basis, to make sure the user catalogs that should be specified for the COPY step have not changed.
2. **Shut down or suspend the source DB2 subsystem.**
 - If an offline clone is desired (DB2 is shut down), choose one of the offline cloning procedures in Chapter 9, "Cloning DB2 subsystems," on page 95 and complete that procedure prior to beginning Step 3.
 - If an online clone is desired (DB2 is suspended), choose one of the online cloning procedures in Chapter 9, "Cloning DB2 subsystems," on page 95 and complete that procedure prior to beginning Step 3.

Note: If you are cloning with consistent FlashCopy, SnapShot, or TimeFinder/Clone, you do not need to suspend the source DB2 subsystem.

These instructions are a prerequisite to Step 3.

3. **COPY step.** The COPY step initiates volume copies, and concurrently backs up the source ICF catalog data relevant to the data sets on the source volumes being copied. The COPY step completes when copy initiations are complete and all ICF catalog data has been backed up. Source ICF catalogs can also be cloned and read from the target volume. See the COPY command for more information.
4. **Start or resume the source DB2 subsystem.**

Note: This step is not necessary if using consistent FlashCopy, SnapShot, or TimeFinder/Clone.

Access, including modification to source volumes, may be resumed after the COPY step completes successfully. However, if time is not of the essence regarding source volume access, or if the RENAME step runs in a relatively short time frame, it is recommended that source volume access resume only after the RENAME step has completed.

Errors such as volumes not specified that should have been, or catalogs incorrectly specified, are not detected until the RENAME step. These categories of errors require the COPY step to be rerun. If the source volumes have changed since the first running of the COPY step, the same P.I.T. (point-in-time) images cannot be captured.

A typical implementation of the COPY step is to use it as a trigger for an application requiring access to source volumes. This can be accomplished by positioning the COPY job as a predecessor to the application, via a job scheduler.

- If an offline clone was done, start the source DB2 subsystem up. See Chapter 9, “Cloning DB2 subsystems,” on page 95.
 - If an online clone was done, resume the source DB2 subsystem. See Chapter 9, “Cloning DB2 subsystems,” on page 95.
5. **RENAME step.** The RENAME step renames and catalogs target volume data sets. The SAFE option, a keyword of the RENAME command is recommended so the RENAME step can be rerun to correct problems caused by mistakes such as incorrectly specified rename masks.

Because of DB2 naming standard requirements, only the DB2 logs and BSDSs can be renamed beyond the hlq. All other DB2 files can only be renamed at the hlq:

- DB2 directory – DB2 expects a specific naming standard:
*hlq.DSNDBx.DSNDB01.**
 - DB2 catalog – DB2 expects a specific naming standard:
*hlq.DSNDBx.DSNDB06.**
 - DB2 databases – DB2 expects a specific naming standard
vcat.DSNDBx.dbname.psname.y0001.Annn
6. **DB2 cloning procedures.** Perform one of the cloning procedures found in Chapter 9, “Cloning DB2 subsystems,” on page 95 before beginning Step 7.
7. **BCSCLEAN step (optional).** Assuming the DB2 subsystem clone is repetitive, each cycle of the process will likely leave orphaned catalog entries for data sets not used in subsequent cycles of the application.

If the target catalog(s) is dedicated to just the target volume data sets, this problem can be avoided by simply placing an IDCAMS step before the COPY step to delete and redefine the target catalog(s) or use BCSCLEAN to delete ICF catalog entries for data sets DB2 Cloning Tool knows were created during the last cycle. If the redefined ICF catalog is not on the same volume it was on prior to the delete, special care must be taken to inform all the ICF catalog address spaces of its new location. IBM informational APAR II13354 details the steps necessary to ensure all sharing systems can access the ICF catalog.

If the target ICF catalog is used for data sets other than those on target volumes, the BCSCLEAN function will delete ICF catalog entries for data sets DB2 Cloning Tool knows were created during the last cycle. This step can be placed anywhere between the end of target volume access and the next start of the COPY process.

Chapter 8. Setting up to copy by volume with an onsite mirror tool

Follow this procedure if clones are created by an onsite mirror tool that does an establish, waits for the data to be copied to the target volume, and then splits the mirror relationship at the appropriate point in time. This could include user-created clones and clones created with IBM PPRC, EMC TimeFinder/Mirror, Hitachi ShadowImage, Softek Replicator, and Innovation Data Processing FDRPAS.

If volume copies are to be created with FlashCopy or SnapShot via DFSMSdss, or TimeFinder/Clone, see Chapter 7, “Setting up to copy by volume with FlashCopy, SnapShot, or TimeFinder/Clone,” on page 83.

The steps to accomplish the DB2 Cloning Tool function are placed within the application's job stream, typically in multiple locations, in order for the function steps to run at the appropriate times, relative to the application's activities. The DB2 Cloning Tool steps are invoked via commands to a common program.

If multiple commands are supplied within the same step, any command that results in a return code equal to or greater than eight will terminate the step. For instance, if COPY and RENAME are in the same step, and COPY terminates with a return code 8, because insufficient target volumes were found, RENAME will be bypassed. Several commands specific to cloning a DB subsystem are not mentioned here. See Chapter 9, “Cloning DB2 subsystems,” on page 95 for more information.

The available commands are:

FINDUCATS

Locates ICF user catalogs pointing to source volume data sets.

COPY Captures ICF catalog data, and optionally, clips and varies target volumes online.

RENAME

Renames and catalogs target volume data sets.

BCSCLEAN

Deletes catalog entries from previous executions.

DB2 Cloning Tool process for user-created target volume clones

The table that follows summarizes the steps required for user-created clones.

Table 16. DB2 Cloning Tool process for user-created target volume clones

Step	Description
1	FINDUCATS step. Optional step to run one time or occasionally, to find user catalogs pointing to source volume data sets.
2	User-supplied step to establish continual mirrors.
3	Shut down or suspend the source DB2 subsystem. See Chapter 9, “Cloning DB2 subsystems,” on page 95 for additional instructions. These instructions are a prerequisite to Step 4. This step is not necessary if using consistent split or break mirror.
4	User-supplied step to split or break mirrors.

Table 16. DB2 Cloning Tool process for user-created target volume clones (continued)

Step	Description
5	COPY step. This step backs up the source ICF catalogs that point to the source volume data sets in synchronization with the split. Source ICF catalogs can also be cloned and read off the target volumes. See the COPY command for more information.
6	Start up or resume the source DB2 subsystem. See Chapter 9, "Cloning DB2 subsystems," on page 95 for additional instructions. These instructions are a prerequisite to Step 7. This step is not necessary if using consistent split or break mirror.
7	RENAME step. This step renames and catalogs target volume data sets.
8	DB2 Cloning procedure. See Chapter 9, "Cloning DB2 subsystems," on page 95 for additional instructions.
9	BCSCLEAN step. Optional step runs after target volume usage is terminated and before the next DB2 Cloning Tool COPY.

Steps for user-created target volumes

These are the steps user-created target volumes.

Procedure

1. **FINDUCATS step (optional).** FINDUCATS identifies which user ICF catalogs contain entries for data sets on the source volumes to be copied. The COPY step requires pairs of source/target user ICF catalogs to be specified. FINDUCATS does not negate this need. It is intended to be run prior to initial setup, and possibly on an occasional basis, to make sure the user catalogs that should be specified for the COPY step have not changed.
2. **User-supplied step to establish continual mirrors.** This step must be supplied by the user and placed in the application stream, sufficiently ahead of the desired 'split time', such that the target volumes will be synchronized with their source volume counterparts.
3. **Shut down or suspend the source DB2 subsystem.**
 - If an offline clone is desired (DB2 is shut down), choose one of the offline cloning procedures in Chapter 9, "Cloning DB2 subsystems," on page 95 and complete that procedure prior to beginning Step 4.
 - If an online clone is desired (DB2 is suspended), choose one of the online cloning procedures in Chapter 9, "Cloning DB2 subsystems," on page 95 and complete that procedure prior to beginning Step 4.

Note: If you are cloning with consistent split or break mirror, you do not need to suspend the source DB2 subsystem.

These instructions are a prerequisite to Step 4.

4. **User-supplied step to split or break mirrors.** This step must be supplied by the user to split mirrors created in step 2. For Hitachi ShadowImage, when using the ICKDSF PPRCOPY commands, the pairs must be suspended with a steady split request before deleting the pairs. The suspend with steady split request causes any pending updates to the target volume to be externalized.
5. **COPY step.** The COPY step in this scenario is issued with the 'DATA-MOVER(PGM(NONE))' parameter. 'DATA-MOVER(PGM(NONE))' implies that target volumes have already been created. The COPY command in this case only backs up the source ICF catalog information needed to rename

and catalog the target volume data sets. Optionally, it relabels the target volumes and varies them online for RENAME processing. The COPY should be done immediately after the SPLIT. Source ICF catalogs can also be cloned and read from the target volume. See the COPY command for more information.

6. **Start or resume the source DB2 subsystem.**

Note: This step is not necessary if you are cloning using consistent split or break mirror.

Access, including modification to source volumes, may be resumed after the COPY step completes successfully. However, if time is not of the essence regarding source volume access, or if the RENAME step runs in a relatively short time frame, it is recommended that source volume access resume only after the RENAME step has completed.

Errors such as volumes not specified that should have been, or catalogs incorrectly specified, are not detected until the RENAME step. These categories of errors require the COPY step to be rerun. If the source volumes have changed since the first running of the COPY step, the same P.I.T. (point-in-time) images cannot be captured.

A typical implementation of the COPY step is to use it as a trigger for an application requiring access to source volumes. This can be accomplished by positioning the COPY job as a predecessor to the application, via a job scheduler.

- If an offline clone was done, start the source DB2 subsystem up. See Chapter 9, "Cloning DB2 subsystems," on page 95.
- If an online clone was done, resume the source DB2 subsystem. See Chapter 9, "Cloning DB2 subsystems," on page 95.

7. **RENAME step.** The RENAME step renames and catalogs target volume data sets. The SAFE option, a keyword of the RENAME command is recommended so the RENAME step can be rerun to correct problems caused by mistakes such as incorrectly specified rename masks.

Because of DB2 naming standard requirements, only the DB2 logs and BSDSs can be renamed beyond the hlq. All other DB2 files expect the following:

- DB2 directory – DB2 expects a specific naming standard:
*hlq.DSNDBx.DSNDB01.**
- DB2 catalog – DB2 expects a specific naming standard:
*hlq.DSNDBx.DSNDB06.**
- DB2 databases – DB2 expects a specific naming standard:
vcat.DSNDBx.dbname.psname.y0001.Annn

8. **DB2 cloning procedures.** Perform one of the cloning procedures found in Chapter 9, "Cloning DB2 subsystems," on page 95 before beginning Step 9.

9. **BCSCLEAN step (optional).** Assuming the DB2 subsystem clone is repetitive, each cycle of the process will likely leave orphaned ICF catalog entries for data sets not used in subsequent cycles of the application.

If the target ICF catalog(s) is dedicated to just the target volume data sets, this problem can be avoided by simply placing an IDCAMS step before the COPY step to delete and redefine the target ICF catalog(s) or use BCSCLEAN to delete catalog entries for data sets DB2 Cloning Tool knows were created during the last cycle. If the redefined ICF catalog is not on the same volume it was on prior to the delete, special care must be taken to inform all the ICF catalog address spaces of its new location. IBM informational APAR II13354 details the steps necessary to ensure all sharing systems can access the catalog.

If the target ICF catalog is used for data sets other than those on target volumes, the BCSCLEAN function will delete catalog entries for data sets DB2 Cloning Tool knows were created during the last cycle. This step can be placed anywhere between the end of target volume access and the next start of the COPY process.

DB2 Cloning Tool process for EMC TimeFinder/Mirror target volume clones

The following table summarizes the steps to accomplish the EMC TimeFinder/Mirror ESTABLISH, SPLIT, and RE-ESTABLISH. More detail is provided in successive topics. For more information, see the EMC TimeFinder Product Guide.

Table 17. DB2 Cloning Tool process for EMC TimeFinder/Mirror target volume clones

Step	Description
1	FINDUCATS step. Optional step to run one time or occasionally, to find user catalogs pointing to source volume data sets.
2	EMC ESTABLISH step. This step establishes the BCV mirrors. EMC RE-ESTABLISH step. Once the mirror relationship has been established and split, the re-establish will synchronize a previously split pair.
3	Shut down or suspend the source DB2 subsystem. See Chapter 9, "Cloning DB2 subsystems," on page 95 for additional instructions. These instructions are a prerequisite to Step 4.
4	EMC SPLIT step. This step splits the mirrors or BCVs.
5	COPY step. This step backs up the source ICF catalogs that point to the source volume data sets in synchronization with the split. Source ICF catalogs can also be cloned and read off the target volumes. See the COPY command for more information.
6	Start up or resume the source DB2 subsystem. See Chapter 9, "Cloning DB2 subsystems," on page 95 for additional instructions. These instructions are a prerequisite to Step 7.
7	RENAME step. This step renames and catalogs target volume data sets.
8	DB2 Cloning procedure. See Chapter 9, "Cloning DB2 subsystems," on page 95 for additional instructions.
9	BCSCLEAN step. Optional step runs after target volume usage is terminated and before the next COPY.

Steps for EMC TimeFinder/Mirror target volume clones

These are the steps to accomplish the EMC TimeFinder/Mirror ESTABLISH, SPLIT, and RE-ESTABLISH.

Procedure

1. **FINDUCATS step (optional).** FINDUCATS identifies which user ICF catalogs contain entries for data sets on the source volumes to be copied. The COPY step requires pairs of source/target user ICF catalogs to be specified. FINDUCATS does not negate this need. It is intended to be run prior to initial setup, and possibly on an occasional basis, to make sure the user catalogs that should be specified for the COPY step have not changed.

2. **User-supplied step to establish continual mirrors.** This step establishes the mirror relationship between the source and target volumes. It must be placed in the application stream, sufficiently ahead of the desired 'split time', such that the target volumes will be synchronized with their source volume counterparts. If this process is repetitive for each cycle, the TimeFinder/Mirror RE-ESTABLISH can be used for subsequent synchronization. The WAIT parameter waits until the BCVs and the standards are synchronized before the job completes. For more information, see the EMC TimeFinder Product Guide.

The following is a sample of an EMC TimeFinder/Mirror ESTABLISH.

```
//STEP1 EXEC PGM=EMCTF
//STEPLIB DD DISP=SHR,DSN=h1q.EMC.LINKLIB
//SYSUDUMP DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SYSIN DD *
GLOBAL MAXRC=4,WAIT
ESTABLISH 01,4120-4121,4100-4101
ESTABLISH 01,4130,4110
//*
```

The following is a sample of an EMC TimeFinder/Mirror RE-ESTABLISH.

```
//STEP1 EXEC PGM=EMCTF
//STEPLIB DD DISP=SHR,DSN=h1q.EMC.LINKLIB
//SYSUDUMP DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SYSIN DD *
GLOBAL MAXRC=4,WAIT
RE-ESTABLISH 01,4120-4121
RE-ESTABLISH 01,4130
//*
```

3. **Shut down or suspend the source DB2 subsystem.**

- If an offline clone is desired (DB2 is shut down), choose one of the offline cloning procedures in Chapter 9, "Cloning DB2 subsystems," on page 95 and complete that procedure prior to beginning Step 4.
- If an online clone is desired (DB2 is suspended), choose one of the online cloning procedures in Chapter 9, "Cloning DB2 subsystems," on page 95 and complete that procedure prior to beginning Step 4.

These instructions are a prerequisite to Step 4.

4. **TimeFinder step to split the mirror relationships.** This step is required to split the mirror relationships to enable processing against the target volumes. The WAIT parameter can elongate the split because it waits until the background process is complete. This would be required if a TimeFinder RESTORE operation is done after the split, but in the case of a volume relabel and data set renames, the NOWAIT parameter is sufficient.

The following is a sample EMC TimeFinder/Consistency Groups consistent SPLIT.

```
//STEP2 EXEC PGM=EMCTF
//STEPLIB DD DISP=SHR,DSN=SYMMI.EMC.TF510.LINKLIB
//SYSUDUMP DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SYSIN DD *
GLOBAL MAXRC=4,NOWAIT
SPLIT 01,4120-4121,CONS(LOCAL(BYP))
SPLIT 01,4130,CONS(LOCAL(BYP))
//*
```

5. **COPY step.** The COPY step in this scenario is issued with the 'DATA-MOVER(PGM(NONE))' parameter. 'DATA-MOVER(PGM(NONE))' implies that target volumes have already been created. COPY in this case only backs up the source ICF catalog information needed to rename and catalog the

target volume data sets. Optionally, it relabels the target volumes and varies them online for RENAME processing. The COPY should be done immediately after the SPLIT.

The following is a sample COPY command for EMC TimeFinder/Mirror BCVs.

```
//STEP3 EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DSN=hlq.SCKZLOAD,DISP=SHR
//CKZINI DD DSN=hlq.SCKZPARM(CKZINI),DISP=SHR
//CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//JOURNAL DD DSN=PRD.DB2A.JRNL,RECOG=KS,KEYLEN=64,KEYOFF=0,DISP=(,CATLG),
// UNIT=SYSALLDA,LRECL=600,SPACE=(CYL,(10,10))
//CKZIN DD *
COPY DATA-MOVER(PGM(NONE)) -
VOLPAIRSDEVN(DB2A01 DB2B01 4120 -
DB2A02 DB2B02 4121 -
DB2A03 DB2B03 4130) -
USERCATALOGS(SRC.USERCAT1 TGT.USERCAT1) -
CATWORK-DSN(PRD.DB2A.WRK.*) -
JOURNAL-DDN(JOURNAL)
//*
```

6. **Start or resume the source DB2 subsystem.** Access, including modification to source volumes, may be resumed after the COPY step completes successfully. However, if time is not of the essence regarding source volume access, or if the RENAME step runs in a relatively short time frame, it is recommended that source volume access resume only after the RENAME step has completed.

Errors such as volumes not specified that should have been, or catalogs incorrectly specified, are not detected until the RENAME step. These categories of errors require the COPY step to be rerun. If the source volumes have changed since the first running of the COPY step, the same P.I.T. (point-in-time) images cannot be captured.

A typical implementation of the COPY step is to use it as a trigger for an application requiring access to source volumes. This can be accomplished by positioning the COPY job as a predecessor to the application, via a Job Scheduler.

- If an offline clone was done, start the source DB2 subsystem up. See Chapter 9, “Cloning DB2 subsystems,” on page 95.
 - If an online clone was done, resume the source DB2 subsystem. See Chapter 9, “Cloning DB2 subsystems,” on page 95.
7. **RENAME step.** The RENAME step renames and catalogs target volume data sets. The SAFE option, a keyword of the RENAME command is recommended so the RENAME step can be rerun to correct problems caused by mistakes such as incorrectly specified rename masks.

Because of DB2 naming standard requirements, only the DB2 logs and BSDSs can be renamed beyond the hlq. All other DB2 files expect the following:

- DB2 directory – DB2 expects a specific naming standard:
*hlq.DSNDBx.DSNDB01.**
 - DB2 catalog – DB2 expects a specific naming standard:
*hlq.DSNDBx.DSNDB06.**
 - DB2 databases – DB2 expects a specific naming standard:
vcat.DSNDBx.dbname.psname.y0001.Annn
8. **DB2 cloning procedures.** Perform one of the cloning procedures found in Chapter 9, “Cloning DB2 subsystems,” on page 95 before beginning Step 9.
 9. **BCSCLEAN step (optional).** Assuming the DB2 subsystem clone is repetitive, each cycle of the process will likely leave orphaned ICF catalog entries for data sets not used in subsequent cycles of the application.

If the target ICF catalog(s) is dedicated to just the target volume data sets, this problem can be avoided by simply placing an IDCAMS step before the COPY step to delete and redefine the target ICF catalog(s) or use BCSCLEAN to delete catalog entries for data sets DB2 Cloning Tool knows were created during the last cycle. If the redefined ICF catalog is not on the same volume it was on prior to the delete, special care must be taken to inform all the ICF catalog address spaces of its new location. IBM informational APAR II13354 details the steps necessary to ensure all sharing systems can access the catalog.

If the target ICF catalog is used for data sets other than those on target volumes, the BCSCLEAN function will delete catalog entries for data sets DB2 Cloning Tool knows were created during the last cycle. This step can be placed anywhere between the end of target volume access and the next start of the COPY process.

Chapter 9. Cloning DB2 subsystems

Separate DB2 cloning procedures by volume are provided for various situations. Choose the appropriate procedure for your situation.

DB2 offline cloning procedures

An offline cloning occurs when DB2 is shut down to clone the data.

- If an offline clone is desired and a DB2 subsystem is being cloned with the intent of a second DB2 subsystem accessing the renamed data sets, refer to the instructions in the topic “DB2 offline cloning procedure” on page 96.
- If an offline clone of a data sharing group with removal of members is desired, refer to the instructions in the topic “DB2 offline cloning with removal of data sharing members procedure” on page 98.
- If an offline clone of a data sharing group with the target being non-data sharing is desired, refer to the instructions in the topic “DB2 offline cloning with target becoming non-data sharing procedure” on page 101.

DB2 online cloning procedures

An online cloning occurs when DB2 SET LOG SUSPEND is issued to suspend transactions, or when a consistent copy process is used such as consistent FlashCopy, SnapShot, or TimeFinder/Clone, or consistent split or break mirror.

- If an online clone is desired, refer to the instructions in the topic “DB2 online cloning procedure” on page 105.
- If an online clone of a data sharing group with removal of members is desired, refer to the instructions in the topic “DB2 online cloning with removal of data sharing members procedure” on page 111.
- If an online clone of a data sharing group with the target being non data sharing is desired, refer to the instructions in the topic “DB2 online cloning with target becoming non-data sharing procedure” on page 117.

DB2 offline cloning

An offline DB2 subsystem clone is created by stopping the source DB2 subsystem to achieve your point-in-time copy. Stopping the source DB2 subsystem ensures that all buffers have been flushed, all data has been committed to disk, and that no transactions are in flight.

| **Important:** If utilities might be running or registered in SYSUTILX on the source
| DB2 subsystem when it is cloned, SYSUTILX and its indexes should be cleaned out
| when DB2 conditioning is complete. You can use the DB2UTILXCLEAN command
| for this purpose. If SYSUTILX is not cleaned out, the source DB2 subsystem might
| become corrupted when the target DB2 subsystem issues a **-TERM UTIL** command
| to clean up the entries it finds in SYSUTILX.

If the DB2 system is data sharing and the DDF LOCATION is being changed, when the target DB2 systems are started they might issue the message:

```
DSNJ707E LOCATION NAME location-name IN BSDS DOES NOT MATCH THE  
LOCATION NAME location-name ASSOCIATED WITH THE DATA SHARING GROUP.
```

This message does not indicate a problem. The message is issued due to the way DB2 rebuilds the SCA when the target DB2 system is initially started.

DB2 offline cloning procedure

Use this procedure if an offline DB2 subsystem is being cloned (when the source DB2 subsystem has been stopped to achieve your point-in-time copy) with the intent of a second or target DB2 subsystem accessing the renamed data sets.

Important: If utilities might be running or registered in SYSUTILX on the source DB2 subsystem when it is cloned, SYSUTILX and its indexes should be cleaned out when DB2 conditioning is complete. You can use the DB2UTILXCLEAN command for this purpose. If SYSUTILX is not cleaned out, the source DB2 subsystem might become corrupted when the target DB2 subsystem issues a **-TERM UTIL** command to clean up the entries it finds in SYSUTILX.

A target DB2 subsystem will use the same buffer pool specifications as its corresponding source DB2 subsystem. If the buffer pool definitions in the source DB2 subsystem are large, care should be taken that sufficient real and auxiliary storage exists to support the size of the buffer pools in the target DB2 subsystem until ALTER BUFFERPOOL commands can be issued.

The names of the work databases are not changed as part of the cloning. The target DB2 system will use the same database names as the source DB2 system. For data sharing, if you want the work database names in the target DB2 system to include a target member identifier, the work databases will need to be manually dropped and created with the desired names.

Table 18. DB2 offline cloning procedure

Step	Description
1	Run the DB2STOP command, SCKZJCL member (CKZDSTO).
2	Run the COPY command, SCKZJCL member (CKZCOPY).
3	Run the DB2START command, SCKZJCL member (CKZDSTA).
4	Run the RENAME command, SCKZJCL member (CKZREN).
5	Run the DB2UPDATE command, SCKZJCL member (CKZDUPD).
6	If DB2 data sharing is used, run the DB2UPDATE command again, SCKZJCL member (CKZDUPD2).
7	Run the DB2START command, SCKZJCL member (CKZDSTA).
8	Run the DB2SQL command, SCKZJCL member (CKZDSQL).
9	Run the DB2STOP command, SCKZJCL member (CKZDSTO).
10	(Optional) Run the DB2UTILXCLEAN command, SCKZJCL member (CKZDUTCL).
11	Run the DB2START command, SCKZJCL member (CKZDSTA).

Steps for DB2 offline cloning

These are the steps for DB2 offline cloning.

Procedure

1. **Run the DB2STOP command.** Run the DB2STOP command using SCKZJCL member (CKZDSTO) to stop the source DB2 subsystem. This ensures that buffers have been flushed, all data has been committed to disk, and no transactions are in flight.

2. **Run the COPY command.** Copy the source volumes to the target volumes and back up the source ICF catalogs that point to the data sets on the source volumes being copied.

If you are using FlashCopy, SnapShot, or TimeFinder/Clone, run the COPY command using SCKZJCL member (CKZCOPY) to copy the source volumes to the target volumes and back up the source ICF catalogs that point to the data sets on the source volumes being copied. See Chapter 23, "DB2 Cloning Tool Subsystem Cloning commands," on page 337 for more information on the COPY command.

If you are using a 'Split of a Continuous Mirror' tool, this step assumes the mirror relationship has been started in advance of when you want to 'split or suspend' the mirror relationship. Issue the 'split or suspend', then run the COPY command using SCKZJCL member (CKZCOPY) to back up the source ICF catalogs that point to the data sets on the source volumes being copied, and optionally, relabel and vary online the target volumes.

3. **Run the DB2START command.** Run the DB2START command using SCKZJCL member (CKZDSTA) to start the source DB2 subsystem. At this point, the data has been cloned and the source volumes are no longer required.
4. **Run the RENAME command.** Run the RENAME command using SCKZJCL member (CKZREN) to rename and catalog the target volume data sets.
5. **Run the DB2UPDATE command.** Run the DB2UPDATE command using SCKZJCL member (CKZDUPD). The DB2UPDATE command makes the necessary DB2 changes to reflect the renamed data sets. DB2UPDATE updates the DB2 directory and the DB2 boot strap data sets (BSDSs).
 - DB2 directory updates – the VCATNAME, and optionally, the DB2 storage group names are updated.
 - BSDS updates – the DB2 catalog name, and 'active' log data set names are updated. Optionally, the ARCHIVE data set names and volume serial numbers in the BSDS are updated.

If data sharing is used in your environment, the target DB2 XCF structures need to be deallocated. The DB2-XCFCLEAN(Y) keyword, the default, will be used to instruct DB2UPDATE to deallocate the target DB2 XCF structures. If this keyword is set to (N), the target DB2 XCF structures need to be manually deallocated prior to starting the target DB2 subsystem.

6. **If DB2 data sharing is used, run DB2UPDATE again.** If DB2 data sharing is used in your environment, run the DB2UPDATE command again, this time using SCKZJCL member (CKZDUPD2) for each additional data sharing member. This step is in addition to the DB2UPDATE command SCKZJCL member (CKZDUPD) previously executed. The DB2UPDATE BSDSONLY command using SCKZJCL member (CKZDUPD2) makes the same changes to each subsequent member's BSDSs as the DB2UPDATE command in the previous step. However, CKZDUPD2 does not do any DB2 directory updates.
7. **Run the DB2START command.** To start the target DB2 subsystem in maintenance mode using the special zparms, run the DB2START command. Use the SPECIAL and DSNZPARM keywords on the target subsystem using SCKZJCL member (CKZDSTA).
8. **Run the DB2SQL command to update the DB2 catalog.** This step assumes that the plan and package have already been installed.

Run the DB2SQL command on the target subsystem using SCKZJCL member (CKZDSQL). The DB2SQL command makes the necessary changes to the DB2 catalog. The changes include the VCATNAME, storage group names, and volumes.

9. **Run the DB2STOP command.** Run the DB2STOP command on the target subsystem using SCKZJCL member (CKZDSTO) to stop the target DB2 subsystem from running in maintenance mode.
10. **(Optional) Run the DB2UTILXCLEAN command.** To clean out all information in SYSUTILX, run the DB2UTILXCLEAN command using SCKZJCL member (CKZDUTCL). If utilities may have been running or registered in SYSUTILX when the source DB2 subsystem is cloned, SYSUTILX and its indexes should be cleaned out. There might be table and index spaces that have UT status due to utilities that were running or registered in SYSUTILX when the source DB2 subsystem was cloned.
11. **Run the DB2START command.** Run the DB2START command using SCKZJCL member (CKZDSTA) to start the target DB2 subsystem again, with its normal zparms DSNZPARx, whenever you are ready to resume application access to the target volumes.

DB2 offline cloning with removal of data sharing members procedure

DB2 offline cloning infers that the source DB2 subsystem has been stopped to achieve your point-in-time copy. It ensures that all buffers have been flushed, all data has been committed to disk, and that no transactions are in flight.

Important: If utilities might be running or registered in SYSUTILX on the source DB2 subsystem when it is cloned, SYSUTILX and its indexes should be cleaned out when DB2 conditioning is complete. You can use the DB2UTILXCLEAN command for this purpose. If SYSUTILX is not cleaned out, the source DB2 subsystem might become corrupted when the target DB2 subsystem issues a **-TERM UTIL** command to clean up the entries it finds in SYSUTILX.

This procedure requires the following:

- The new target DB2 data sharing group members will have no log history, so image copies must be taken if further recovery is desired.
- For a data sharing environment, the target DB2 XCF structures should be deallocated prior to the first starting of the target DB2 subsystem.
- Only the DB2 members being retained need to be cloned.
- The RENAME-MASKS keyword in the RENAME command needs to include entries that will cause the renaming of all the member BSDSs.
- The names of the work databases are not changed as part of the cloning. The target DB2 system will use the same database names as the source DB2 system. For data sharing, if you want the work database names in the target DB2 system to include a target member identifier, the work databases will need to be manually dropped and created with the desired names.

The following table provides the steps for offline cloning with removal of data sharing members:

Table 19. DB2 offline cloning with removal of data sharing members procedure

Step	Description
1	Run the DB2STOP command, SCKZJCL member (CKZDSTO).
2	Run the COPY command, SCKZJCL member (CKZCOPY).
3	Run the DB2START command, SCKZJCL member (CKZDSTA).
4	Run the RENAME command, SCKZJCL member (CKZREN).

Table 19. DB2 offline cloning with removal of data sharing members procedure (continued)

Step	Description
5	Run the DB2UPDATE command, SCKZJCL member (CKZDUPD).
6	Run the DB2UPDATE command again, SCKZJCL member (CKZDUPD2).
7	Run the DB2RBLDBSDS command, SCKZJCL member (CKZDRBBS).
8	Run the DB2LGRNXCLEAN command, SCKZJCL member (CKZDLGCL).
9	Run the DB2START command, SCKZJCL member (CKZDSTA).
10	Run the DB2SQL command, SCKZJCL member (CKZDSQL).
11	Set the DB2 buffer pool attributes.
12	Run the DB2STOP command, SCKZJCL member (CKZDSTO).
13	(Optional) Run the DB2UTILXCLEAN command, SCKZJCL member (CKZDUTCL).
14	Run the DB2START command, SCKZJCL member (CKZDSTA), to start the target DB2 subsystem again, using its normal zparms DSNZPARx.
15	Run the DB2START command, SCKZJCL member (CKZDSTA), to start the other target DB2 subsystems, using their normal zparms DSNZPARx.
16	Set the DB2 buffer pool attributes.

Steps for DB2 offline cloning with removal of data sharing members

These are the steps for DB2 offline cloning with removal of data sharing members.

Procedure

1. **Run the DB2STOP command.** Run the DB2STOP command using SCKZJCL member (CKZDSTO) to stop the source DB2 subsystem. This ensures that buffers have been flushed, all data has been committed to disk, and no transactions are in flight.
2. **Run the COPY command.** Copy the source volumes to the target volumes and back up the source ICF catalogs that point to the data sets on the source volumes being copied.

If you are using IBM FlashCopy or SnapShot, run the COPY command using SCKZJCL member (CKZCOPY) to copy the source volumes to the target volumes and back up the source ICF catalogs that point to the data sets on the source volumes being copied. See Chapter 23, “DB2 Cloning Tool Subsystem Cloning commands,” on page 337 for more information on the COPY command.

If you are using a 'Split of a Continuous Mirror' tool, this step assumes the mirror relationship has been started in advance of when you want to 'split or suspend' the mirror relationship. Issue the 'split or suspend', then run the COPY command using SCKZJCL member (CKZCOPY) to back up the source ICF catalogs that point to the data sets on the source volumes being copied, and optionally, relabel and vary online the target volumes.

3. **Run the DB2START command.** Run the DB2START command on the source system using SCKZJCL member (CKZDSTA). At this point, the data has been cloned and the source volumes are no longer required.
4. **Run the RENAME command.** Run the RENAME command using SCKZJCL member (CKZREN) to rename and catalog the target volume data sets.

5. **Run the DB2UPDATE command.** Run DB2UPDATE command using SCKZJCL member (CKZDUPD). The source DB2 is data sharing so the DB2-GROUP and DB2-MEMBERS keywords must be used.

The DB2UPDATE command makes the necessary DB2 changes to reflect the renamed data sets. DB2UPDATE updates the DB2 directory and the DB2 boot strap data sets (BSDSs).

- DB2 directory updates – the VCATNAME, and optionally, the DB2 storage group names are updated.
- BSDS updates – the DB2 catalog name, and 'active' log data set names are updated. Optionally, the ARCHIVE data set names and volume serial numbers in the BSDS are updated.

The target DB2 XCF structures and group members need to be deallocated. The DB2-XCF CLEAN(Y) keyword, the default, will be used to instruct DB2UPDATE to deallocate the target DB2 XCF structures. If this keyword is set to (N), the target DB2 XCF structures need to be manually deallocated prior to starting the target DB2 subsystem.

6. **Run the DB2UPDATE command again.** Run the DB2UPDATE command again, this time using SCKZJCL member (CKZDUPD2) for each additional data sharing member that is being cloned. This step is in addition to the DB2UPDATE command SCKZJCL member (CKZDUPD) previously executed. The DB2UPDATE BSDSONLY command using SCKZJCL member (CKZDUPD2) makes the same changes to each subsequent member's BSDSs as the DB2UPDATE command in the previous step. However, CKZDUPD2 does not do any DB2 directory updates.

The source DB2 is data sharing so the DB2-GROUP and DB2-MEMBERS keywords must be used.

7. **Run the DB2RBLDBSDS command.** Run the DB2RBLDBSDS command using SCKZJCL member (CKZDRBBS) for each data sharing member that is being cloned. This step will rebuild the member BSDS to have only active logs in it.
8. **Run the DB2LGRNXCLEAR command.** Run the DB2LGRNXCLEAR command using SCKZJCL member (CKZDLGCL). This step will clean out all information in SYSLGRNX.
9. **Run the DB2START command.** Run the DB2START command to start the primary target in maintenance mode using the special zparms. Use the SPECIAL and DSNZPARM keywords on the target subsystem using SCKZJCL member (CKZDSTA).

10. **Run the DB2SQL command.** This step assumes that the plan and package have already been installed. For more information on installing the plan and package, see the topic "Cloning a DB2 subsystem" on page 80.

Run the DB2SQL command on the target subsystem using SCKZJCL member (CKZDSQL). The DB2SQL command makes the necessary changes to the DB2 catalog. The changes include the VCATNAME, storage group names, and volumes.

11. **Set the DB2 buffer pool attributes.** The DB2 buffer pool attributes need to be set to the desired values for the primary target DB2 subsystem. The DB2 -ALTER BUFFERPOOL command can be used to set desired buffer pool attributes.
12. **Run the DB2STOP command.** Run the DB2STOP command using SCKZJCL member (CKZDSTO) to stop the primary target DB2 subsystem from running in maintenance mode.
13. **(Optional) Run the DB2UTILXCLEAR command.** To clean out all information in SYSUTILX, run the DB2UTILXCLEAR command using

SCKZJCL member (CKZDUTCL). If utilities may have been running or registered in SYSUTILX when the source DB2 subsystem is cloned, SYSUTILX and its indexes should be cleaned out. There might be table and index spaces that have UT status due to utilities that were running or registered in SYSUTILX when the source DB2 subsystem was cloned.

14. **Run the DB2START command.** Run the DB2START command using SCKZJCL member (CKZDSTA) to start the primary target DB2 subsystem again, with its normal zparms DSNZPARx, whenever you are ready to resume application access to the target volumes.
15. **Run the DB2START command.** Run the DB2START command using SCKZJCL member (CKZDSTA) to start the other target DB2 subsystems that are being cloned, with their normal zparms DSNZPARx, whenever you are ready to resume application access to the target volumes.
16. **Set the DB2 buffer pool attributes.** The DB2 buffer pool attributes need to be set to the desired values for the other target DB2 subsystems that are being cloned. The DB2 -ALTER BUFFERPOOL command can be used to set desired buffer pool attributes.

DB2 offline cloning with target becoming non-data sharing procedure

DB2 offline cloning infers that the source DB2 subsystem has been stopped to achieve your point-in-time copy. It ensures that all buffers have been flushed, all data has been committed to disk, and that no transactions are in flight.

Important: If utilities might be running or registered in SYSUTILX on the source DB2 subsystem when it is cloned, SYSUTILX and its indexes should be cleaned out when DB2 conditioning is complete. You can use the DB2UTILXCLEAN command for this purpose. If SYSUTILX is not cleaned out, the source DB2 subsystem might become corrupted when the target DB2 subsystem issues a **-TERM UTIL** command to clean up the entries it finds in SYSUTILX.

This procedure requires the following:

- As the target DB2 subsystem is to become non-data sharing, both its special and normal zparms need to be non-data sharing.
- For a data sharing environment, the target DB2 XCF structures should be deallocated prior to the first starting of the target DB2 subsystem.
- Only the DB2 member being retained needs to be cloned.
- The RENAME-MASKS keyword in the RENAME command needs to include entries that will cause the renaming of all the member BSDSs.
- A target DB2 subsystem will use the same buffer pool specifications as its corresponding source DB2 subsystem. If the buffer pool definitions in the source DB2 subsystem are large, care should be taken that sufficient real and auxiliary storage exists to support the size of the buffer pools in the target DB2 subsystem until ALTER BUFFERPOOL commands can be issued.
- The names of the work databases are not changed as part of the cloning. The target DB2 system will use the same database names as the source DB2 system. For data sharing, if you want the work database names in the target DB2 system to include a target member identifier, the work databases will need to be manually dropped and created with the desired names.

The following table provides the steps for offline cloning with the target becoming a non-data sharing subsystem:

Table 20. DB2 offline cloning with target becoming non-data sharing procedure

Step	Description
1	Run the DB2STOP command, SCKZJCL member (CKZDSTO).
2	Run the COPY command, SCKZJCL member (CKZCOPY).
3	Run the DB2START command, SCKZJCL member (CKZDSTA).
4	Run the RENAME command, SCKZJCL member (CKZREN).
5	Run the DB2UPDATE command, SCKZJCL member (CKZDUPD).
6	Run the DB2UPDATE command again, SCKZJCL member (CKZDUPD2).
7	Run the DB2RBLDBSDS command, SCKZJCL member (CKZDRBBS).
8	Run the DB2START command, SCKZJCL member (CKZDSTA).
9	Run the DB2SQL command, SCKZJCL member (CKZDSQL), to update the DB2 catalog.
10	Run the DB2STOP command, SCKZJCL member (CKZDSTO).
11	(Optional) Run the DB2UTILXCLEAN command, SCKZJCL member (CKZDUTCL).
12	Run the DB2START command, SCKZJCL member (CKZDSTA).

Steps for DB2 offline cloning with target becoming non-data sharing

These are the steps for DB2 offline cloning with the target becoming non-data sharing.

Procedure

1. **Run the DB2STOP command.** Run the DB2STOP command using SCKZJCL member (CKZDSTO) to stop the source DB2 subsystem. This ensures that buffers have been flushed, all data has been committed to disk, and no transactions are in flight.
2. **Run the COPY command.** Copy the source volumes to the target volumes and back up the source ICF catalogs that point to the data sets on the source volumes being copied.

If you are using FlashCopy, SnapShot, or TimeFinder/Clone, run the COPY command using SCKZJCL member (CKZCOPY) to copy the source volumes to the target volumes and back up the source ICF catalogs that point to the data sets on the source volumes being copied. See Chapter 23, "DB2 Cloning Tool Subsystem Cloning commands," on page 337 for more information on the COPY command.

If you are using a 'Split of a Continuous Mirror' tool, this step assumes the mirror relationship has been started in advance of when you want to 'split or suspend' the mirror relationship. Issue the 'split or suspend', then run the COPY command using SCKZJCL member (CKZCOPY) to back up the source ICF catalogs that point to the data sets on the source volumes being copied, and optionally, relabel and vary online the target volumes.

3. **Run the DB2START command.** Run the DB2START command on the source system using SCKZJCL member (CKZDSTA). At this point, the data has been cloned and the source volumes are no longer required.
4. **Run the RENAME command.** Run the RENAME command using SCKZJCL member (CKZREN) to rename and catalog the target volume data sets. The RENAME-MASKS keyword needs to include entries that will cause the renaming of all the member BSDSs.

5. **Run the DB2UPDATE command.** Run the DB2UPDATE command using SCKZJCL member (CKZDUPD). The source DB2 is data sharing so the DB2-GROUP and DB2-MEMBERS keywords must be used. The DB2UPDATE command makes the necessary DB2 changes to reflect the renamed data sets. DB2UPDATE updates the DB2 directory and the DB2 boot strap data sets (BSDSs).
 - DB2 directory updates – the VCATNAME, and optionally, the DB2 storage group names are updated.
 - BSDS updates – the DB2 catalog name, and ‘active’ log data set names are updated. Optionally, the ARCHIVE data set names and volume serial numbers in the BSDS are updated.

The target DB2 XCF structures and group members need to be deallocated. The DB2-XCF CLEAN(Y) keyword, the default, will be used to instruct DB2UPDATE to deallocate the target DB2 XCF structures. If this keyword is set to (N), the target DB2 XCF structures need to be manually deallocated prior to starting the target DB2 subsystem.

6. **Run the DB2UPDATE command again.** Run the DB2UPDATE command again, this time using SCKZJCL member (CKZDUPD2) for each additional data sharing member. This step is in addition to the DB2UPDATE command SCKZJCL member (CKZDUPD) previously executed. The DB2UPDATE BSDSONLY command using SCKZJCL member (CKZDUPD2) makes the same changes to each subsequent member's BSDSs as the DB2UPDATE command in the previous step. However, CKZDUPD2 does not do any DB2 directory updates.

The source DB2 is data sharing so the DB2-GROUP and DB2-MEMBERS keywords must be used.

7. **Run the DB2RBLDBSDS command.** Run the DB2RBLDBSDS command using SCKZJCL member (CKZDRBBS) for the desired target member using the keyword DATA-SHARING(NO). This step will alter the member BSDS to be non-data sharing.
8. **Run the DB2START command.** Run the DB2START command to start the primary target DB2 subsystem in maintenance mode using the special non-data sharing zparms. Use the SPECIAL, DSNZPARM, and REPLY-TO-RESTART-WTOR(Y) keywords on the target subsystem using SCKZJCL member (CKZDSTA) to start the target DB2 subsystem.

This start-up of the target DB2 subsystem will require a cold start, so an operator reply will need to be made to the DB2 message:

```
DSNJ246I  CONDITIONAL RESTART RECORD INDICATES COLD START AT
RBA xxxxxxxxxxxx.  REPLY Y TO CONTINUE, N TO CANCEL
```

Using the REPLY-TO-RESTART-WTOR(Y) keyword with the DB2START command will cause DB2START to automatically reply to the DSNJ246I WTOR message.

9. **Run the DB2SQL command.** This step assumes that the plan and package have already been installed. For more information on installing the plan and package, see the topic “Cloning a DB2 subsystem” on page 80. Run the DB2SQL command on the target subsystem using SCKZJCL member (CKZDSQL). The DB2SQL command makes the necessary changes to the DB2 catalog. The changes include the VCATNAME, storage group names, and volumes.
10. **Run the DB2STOP command.** Run the DB2STOP command on the target subsystem using SCKZJCL member (CKZDSTO) to stop the target DB2 subsystem from running in maintenance mode.

11. **(Optional) Run the DB2UTILXCLEAN command.** To clean out all information in SYSUTILX, run the DB2UTILXCLEAN command using SCKZJCL member (CKZDUTCL). If utilities may have been running or registered in SYSUTILX when the source DB2 subsystem is cloned, SYSUTILX and its indexes should be cleaned out. There might be table and index spaces that have UT status due to utilities that were running or registered in SYSUTILX when the source DB2 subsystem was cloned.
12. **Run the DB2START command.** Run the DB2START command using SCKZJCL member (CKZDSTA) to start the target DB2 subsystem again, with its normal zparms DSNZPARx, whenever you are ready to resume application access to the target volumes.

DB2 online cloning

An online DB2 subsystem clone is created by suspending the source DB2 subsystem to achieve your point-in-time copy. By suspending the source DB2 subsystem, any pending database writes are forced to disk, update activity is suspended, and the log buffers are flushed to disk. An alternative to suspending the source DB2 subsystem is to use consistent FlashCopy, SnapShot, or TimeFinder/Clone, or consistent split or break mirror to achieve your point-in-time copy.

Important: If utilities might be running or registered in SYSUTILX on the source DB2 subsystem when it is cloned, SYSUTILX and its indexes should be cleaned out when DB2 conditioning is complete. You can use the DB2UTILXCLEAN command for this purpose. If SYSUTILX is not cleaned out, the source DB2 subsystem might become corrupted when the target DB2 subsystem issues a **-TERM UTIL** command to clean up the entries it finds in SYSUTILX.

If the DB2 system is data sharing and the DDF LOCATION is being changed, when the target DB2 systems are started, they might issue the message:

```
DSNJ707E LOCATION NAME location-name IN BSDS DOES NOT MATCH  
THELOCATION NAME location-name ASSOCIATED WITH THE DATA SHARING GROUP.
```

This message does not indicate a problem. The message is issued due to the way DB2 rebuilds the SCA when the target DB2 system is initially started.

Important: In order to prevent source corruption, ensure that for the special zparm DSNZSPEC, the macro DSN6SPRM was changed to use DEFER,ALL. Refer to the topic "Cloning a DB2 subsystem" on page 80 for information about how to set up the special dsnzparm module.

Status of transactions in flight

An online cloning solution often results in transactions in flight. These in-flight transactions, cloned to the target subsystem, result in the same target subsystem action that would happen on the source system if it were to have died at that same time and then restarted. When you use online cloning, the target restart is essentially an emergency restart of a failed system.

The initial starts of the target DB2 subsystems should not be a cold start. A cold start will not allow the target DB2 to resolve in-flight transactions and could leave inconsistencies in DB2 data.

The 'unit of work' or transaction can be in any one of the following states:

In flight

This is where a transaction is most of the time. On the target subsystem it will be backed out to the last commit point. Read-only transactions have nothing to back out.

Commit

The transaction is in the process of taking a commit. On the target subsystem the transaction updates should be committed.

Abort The transaction is in the process of aborting. On the target subsystem it will be backed out to the last commit point

In doubt

The transaction was committing and was between phase 1 and phase 2 commit processing. DB2 does not know if the transaction should be backed out or committed. Manual intervention is required to either back out or commit the transaction.

Log data needed to back out a transaction should be contained in the active logs. It is possible that a back-out will need log data that no longer resides in an active log. In this case, archive logs will be needed to successfully complete the back-out. Whether and which archive logs will be necessary for restart of the target subsystem depends on how busy the source system is and how large the active logs are. We recommend the point-in-time copy created online (via DB2 SET LOG SUSPEND, consistent FlashCopy, SnapShot, or TimeFinder/Clone, or consistent split or break mirror) be done at a quiet time.

If there are DDF transactions active during the cloning, when the target DB2 systems are started they may issue the message:

```
DSNL034E DDF CANNOT BE STARTED BECAUSE OF BSDS INCONSISTENCIES
```

To resolve this condition and allow DDF to start, the following DB2 command must be issued on the target DB2 system:

```
-RESET INDOUBT LUNAME(*) FORCE
```

This DB2 command should not be issued until after the DB2SQL command has been run.

DB2 online cloning procedure

If an online clone is desired, refer to these instructions.

An online clone can be done with one of these methods:

- Using DB2 SET LOG SUSPEND and RESUME to suspend and resume transactions during the cloning process.
- Using consistent FlashCopy, SnapShot, or TimeFinder/Clone to manage I/O to the volumes during the cloning process. The FlashCopy, SnapShot, or TimeFinder/Clone API manages the suspension and resuming of activity against the volumes during cloning.
- Using consistent split or break mirror. The split or break mirror API manages the suspension and resuming of activity against the volumes.

Important: If utilities might be running or registered in SYSUTILX on the source DB2 subsystem when it is cloned, SYSUTILX and its indexes should be cleaned out when DB2 conditioning is complete. You can use the DB2UTILXCLEAN command for this purpose. If SYSUTILX is not cleaned out, the source DB2 subsystem might

become corrupted when the target DB2 subsystem issues a **-TERM UTIL** command to clean up the entries it finds in SYSUTILX.

Important: To prevent source corruption, ensure that for the special zparm DSNZSPEC, the macro DSN6SPRM was changed to use DEFER,ALL. Refer to the topic “Cloning a DB2 subsystem” on page 80 for information on how to set up the special dsnzparm module.

This procedure requires the following:

- If a DB2 data sharing group is being cloned, all members of the group should also be cloned.
- For a data sharing environment, the target DB2 XCF structures should also be deallocated prior to the first starting of the target DB2 subsystem.
- For a DB2 data sharing environment, the special zparms will need to be set up for each target member. For more information, see the topic “Cloning a DB2 subsystem” on page 80.
- Because the source DB2 subsystems are active at the time of the clone, any one of them may have a lock outstanding on a DB2 table. These outstanding locks are carried into the target DB2 sharing environment. To free these locks in the target environment, the corresponding target member will need to be started in maintenance mode with its own special zparms.
- A target DB2 subsystem will use the same buffer pool specifications as its corresponding source DB2 subsystem. If the buffer pool definitions in the source DB2 subsystem are large, care should be taken that sufficient real and auxiliary storage exists to support the size of the buffer pools in the target DB2 subsystem until ALTER BUFFERPOOL commands can be issued.
- The names of the work databases are not changed as part of the cloning. The target DB2 system will use the same database names as the source DB2 system. For data sharing, if you want the work database names in the target DB2 system to include a target member identifier, the work databases will need to be manually dropped and created with the desired names.

The following table provides the steps for online cloning:

Table 21. DB2 online cloning procedure

Step	Description
1	Run the DB2SETLOG command, SCKZJCL member (CKZDSETL). This step is not necessary if using consistent FlashCopy, SnapShot, or TimeFinder/Clone, or consistent split or break mirror.
2	Run the COPY command, SCKZJCL member (CKZCOPY).
3	Run the DB2SETLOG command, SCKZJCL member (CKZDSETL). This step is not necessary if using consistent FlashCopy, SnapShot, or TimeFinder/Clone, or consistent split or break mirror.
4	Run the RENAME command, SCKZJCL member (CKZREN).
5	Run the DB2UPDATE command, SCKZJCL member (CKZDUPD).
6	If DB2 data sharing is used, run the DB2UPDATE command again, SCKZJCL member (CKZDUPD2).
7	Run the DB2START command, SCKZJCL member (CKZDSTA).
8	If DB2 data sharing is used, run the DB2START command, SCKZJCL member (CKZDSTA), to start the remaining target DB2 members.
9	Run the DB2FIX command, SCKZJCL member (CKZDFIX).

Table 21. DB2 online cloning procedure (continued)

Step	Description
	Attention: Steps 10, 11, and 12 are required only if the table space DBD01 (SYSDBDXA for DB2 10 NFM) in database DSNDDB01 was restricted when DB2FIX ran. The changes made to DBD01 (SYSDBDXA for DB2 10 NFM) by DB2UPDATE may have been regressed and must be redone. Otherwise, proceed from step 9 directly to step 13.
10	(Optional, see note.) Run the DB2STOP command, SCKZJCL member (CKZDSTO).
11	(Optional, see note.) Run the DB2UPDATE command again, SCKZJCL member (CKZDUPD).
12	(Optional, see note.) Run the DB2START command, SCKZJCL member (CKZDSTA).
13	Run the DB2SQL command, SCKZJCL member (CKZDSQL).
14	Run the DB2FIX command, SCKZJCL member (CKZDFIX).
15	Run the DB2STOP command, SCKZJCL member (CKZDSTO).
16	(Optional) Run the DB2UTILXCLEAN command, SCKZJCL member (CKZDUTCL).
17	Run the DB2START command, SCKZJCL member (CKZDSTA).

Steps for DB2 online cloning

These are the steps for DB2 online cloning.

Procedure

1. Run the DB2SETLOG command.

Note: This step is not necessary if using consistent FlashCopy, SnapShot, or TimeFinder/Clone, or consistent split or break mirror.

The DB2 Cloning Tool DB2SETLOG command with the SUSPEND keyword issues a SET LOG LOGLOAD(0) command to force any pending database writes to disk, followed by a SET LOG SUSPEND command to suspend update activity and flush the log buffers to disk. On the source system, run the DB2SETLOG command with the SUSPEND keyword, SCKZJCL member (CKZDSETL).

If you are running in a data sharing environment, run the DB2SETLOG command with the SUSPEND keyword, SCKZJCL member (CKZDSETL), for each member in a data sharing environment.

2. Run the COPY command. Copy the source volumes to the target volumes and back up the source ICF catalogs that point to the data sets on the source volumes being copied.

If you are using FlashCopy, SnapShot, or TimeFinder/Clone, run the COPY command using SCKZJCL member (CKZCOPY) to copy the source volumes to the target volumes and back up the source ICF catalogs that point to the data sets on the source volumes being copied. See Chapter 23, "DB2 Cloning Tool Subsystem Cloning commands," on page 337 for more information on the COPY command.

If you are using a 'Split of a Continuous Mirror' tool, this step assumes the mirror relationship has been started in advance of when you want to 'split or suspend' the mirror relationship. Issue the 'split or suspend' then run COPY using SCKZJCL member (CKZCOPY) to back up the source ICF catalogs that point to the data sets on the source volumes being copied.

If a DB2 data sharing group is being cloned, all members of the group should also be cloned.

3. **Run the DB2SETLOG command.**

Note: This step is not necessary if using consistent FlashCopy, SnapShot, or TimeFinder/Clone, or consistent split or break mirror.

After the data has been copied to the target volumes, the source ICF catalogs have been backed up, and COPY has completed successfully, you can resume access to the source volumes.

Run the DB2SETLOG command with the RESUME keyword, SCKZJCL member (CKZDSETL), on the source system to resume update activity. If you are running in a data sharing environment, run the DB2SETLOG command with the RESUME keyword, SCKZJCL member (CKZDSETL), for each member in a data sharing environment.

4. **Run the RENAME command.** Run the RENAME command using SCKZJCL member (CKZREN) to rename and catalog the target volume data sets.

5. **Run the DB2UPDATE command.** Run the DB2UPDATE command using SCKZJCL member (CKZDUPD). The DB2UPDATE command makes the necessary DB2 changes to reflect the renamed data sets. DB2UPDATE updates the DB2 directory and the DB2 boot strap data sets (BSDSs).

- DB2 directory updates – the VCATNAME, and optionally, the DB2 storage group names are updated.
- BSDS updates – the DB2 catalog name, and 'active' log data set names are updated. Optionally, the ARCHIVE data set names and volume serial numbers in the BSDS are updated.

If DB2 data sharing is used in your environment, the target DB2 XCF structures need to be deallocated. The DB2-XCFCLEAN(Y) keyword, the default, will be used to instruct DB2UPDATE to deallocate the target DB2 XCF structures. If this keyword is set to (N), the target DB2 XCF structures need to be manually deallocated prior to starting the target DB2 subsystem. See "Deallocating target DB2 coupling facility structures" on page 591 for more information on this process.

6. **If DB2 data sharing is used, run the DB2UPDATE command again.** If DB2 data sharing is used in your environment, run the DB2UPDATE command again, this time using SCKZJCL member (CKZDUPD2) for each additional data sharing member. This step is in addition to the DB2UPDATE command SCKZJCL member (CKZDUPD) previously executed. The DB2UPDATE command using SCKZJCL member (CKZDUPD2) makes the same changes to each subsequent member's BSDSs as the DB2UPDATE command in the previous step. However, CKZDUPD2 does not do any DB2 directory updates.

7. **Run the DB2START command.**

Important: To prevent source corruption, ensure that for the special zparm DSNZSPEC, the macro DSN6SPRM was changed to use DEFER,ALL. Refer to the topic "Cloning a DB2 subsystem" on page 80 for information on how to set up the special dsnzparm module.

Run the DB2START command to start the target subsystem in maintenance mode using the special zparms. Use the SPECIAL and DSNZPARM keywords on the target subsystem using SCKZJCL member (CKZDSTA).

Note: The initial starts of the target DB2 subsystems should not be a cold start. A cold start will not allow the target DB2 to resolve in-flight transactions and could leave inconsistencies in DB2 data.

The following error messages are expected. If an in-flight transaction was in progress, multiple messages may be associated with it:

```
DSNI001I @TGT1 RESTART HAS BEEN DEFERRED
          REASON 00C90095
          TYPE 00000200
          NAME databasename .tablespace or indexspace name
DSNB250E @TGT1 DSNIMPDA PAGE RANGE WAS ADDED TO THE LOGICAL PAGE LIST
          DATABASE NAME=database name
          SPACE NAME=tablespace or indexspace name
          DATA SET NUMBER=1
          PAGE RANGE X'nnnnnnnn' TO X'nnnnnnnn'
          START LRSN=X'nnnnnnnnnnnn'
          END LRSN=X'nnnnnnnnnnnn'
          START RBA=X'nnnnnnnnnnnn'
```

8. If DB2 data sharing is used, run the DB2START command again.

Important: To prevent source corruption, ensure that for the special zparm DSNZSPEC, the macro DSN6SPRM was changed to use DEFER,ALL. Refer to the topic “Cloning a DB2 subsystem” on page 80 for information on how to set up the special dsnzparm module.

If DB2 data sharing is used in your environment, the other target DB2 members may hold locks that need to be released before the DB2FIX command can be run. To release these locks, start the remaining target DB2 members in maintenance mode with LIGHT(YES) and the special zparms as described in the next paragraph. LIGHT(YES) causes the target DB2 subsystem to stop after it has released any locks it held.

Run the DB2START command with the SPECIAL, LIGHT, and DSNZPARM keywords on the target subsystem using SCKZJCL member (CKZDSTA).

Note: The initial starts of the target DB2 subsystems should not be a cold start. A cold start will not allow the target DB2 to resolve in-flight transactions and could leave inconsistencies in DB2 data.

9. Run the DB2FIX command. Run the DB2FIX command to correct any of the DB2 catalog (DSNDB06) and DB2 Directory (DSNDB01) page spaces that are restricted.

Run the DB2FIX command using SCKZJCL member (CKZDFIX) with the keyword DATABASES(DB2) on the target subsystem. This will start any page spaces in the DB2 catalog (DSNDB06) and DB2 Directory (DSNDB01) that have LPL or GRECP status.

If DB2 data sharing is used in your environment, only one DB2 member should be active when DB2FIX is run.

There may be times when locks are held by other members even though they have been started. The use of MEMBERS-NEED-STARTING(ACTION(CONTINUE)) can be used to cause DB2FIX to issue the START DATABASE commands for table and index spaces in LPL or GRECP status even though other members in the data sharing group have locks outstanding. If a timeout occurs when using MEMBERS-NEED-STARTING(ACTION(CONTINUE)), it may be necessary to run DB2FIX using MEMBERS-NEED-STARTING(ACTION(CONTINUE)) on the other members as well.

Attention: Steps 10, 11, and 12 need to be performed ONLY if the table space DBD01 (SYSDBDXA for DB2 10 NFM) in database DSNDB01 was restricted when DB2FIX ran. The changes made to DBD01 (SYSDBDXA for DB2 10 NFM) by DB2UPDATE may have been regressed and need to be redone. These steps can be automated by using either the DB2FIX return code setting, or the DB2FIX WTO message produced when DB2FIX starts restricted page spaces. For more information, see the “DB2FIX” on page 379 command. Otherwise, proceed to step 13.

10. **Run the DB2STOP command.** Run the DB2STOP command using SCKZJCL member (CKZDSTO) to stop the target DB2 subsystem from running in maintenance mode.
11. **Run the DB2UPDATE command again.** Run the DB2UPDATE command with the DBD01ONLY keyword using SCKZJCL member (CKZDUPD). The DB2UPDATE command makes the necessary DB2 changes to reflect the renamed data sets. DB2UPDATE with the DBD01ONLY keyword updates only the DB2 directory. The VCATNAME, and optionally, the DB2 storage group names are updated.

If DB2 data sharing is used in your environment, the target DB2 XCF structures need to be deallocated. The DB2-XCFCLEAN(Y) keyword, the default, will be used to instruct DB2UPDATE to deallocate the target DB2 XCF structures. If this keyword is set to (N), the target DB2 XCF structures need to be manually deallocated prior to starting the target DB2 subsystem. See “Deallocating target DB2 coupling facility structures” on page 591 for more information on this process.

12. **Run the DB2START command.**

Important: To prevent source corruption, ensure that for the special zparm DSNZSPEC, the macro DSN6SPRM was changed to use DEFER,ALL. Refer to the topic “Cloning a DB2 subsystem” on page 80 for information on how to set up the special dsnzparm module.

Run the DB2START command to start the target DB2 subsystem in maintenance mode using the special zparms. Use the SPECIAL and DSNZPARAM keywords on the target subsystem using SCKZJCL member (CKZDSTA).

13. **Run the DB2SQL command.** Run the DB2SQL command using SCKZJCL member (CKZDSQL) to update the DB2 catalog. This step assumes that the plan and package have already been installed. For more information on installing the plan and package, see the topic “Cloning a DB2 subsystem” on page 80.

The DB2SQL command makes the necessary changes to the DB2 catalog. The changes include the VCATNAME, storage group names, and volumes.

14. **Run the DB2FIX command.** Run the DB2FIX command to correct any of the application page spaces that are restricted. Run the DB2FIX command using SCKZJCL member (CKZDFIX) with the keyword DATABASES(APPLICATION) on the target subsystem. This will start any application page spaces that have LPL or GRECP status.

If DB2 data sharing is used in your environment, only one DB2 member should be active when DB2FIX is run.

There may be times when locks are held by other members even though they have been started. The use of MEMBERS-NEED-STARTING(ACTION(CONTINUE)) can be used to cause DB2FIX to issue the START DATABASE commands for table and index spaces in LPL or GRECP status even though other members in the data sharing group have locks outstanding. If a timeout

occurs when using MEMBERS-NEED-STARTING(ACTION(CONTINUE)), it may be necessary to run DB2FIX using MEMBERS-NEED-STARTING(ACTION(CONTINUE)) on the other members as well.

15. **Run the DB2STOP command.** To stop the target DB2 subsystem from running in maintenance mode, run the DB2STOP command using SCKZJCL member (CKZDSTO).
16. **(Optional) Run the DB2UTILXCLEAN command.** To clean out all information in SYSUTILX, run the DB2UTILXCLEAN command using SCKZJCL member (CKZDUTCL). If utilities may have been running or registered in SYSUTILX when the source DB2 subsystem is cloned, SYSUTILX and its indexes should be cleaned out. There might be table and index spaces that have UT status due to utilities that were running or registered in SYSUTILX when the source DB2 subsystem was cloned.
17. **Run the DB2START command.** To start the target DB2 subsystem using its 'normal' zparms, run the DB2START command using SCKZJCL member (CKZDSTA). You can also start other members of the target data sharing group with their normal zparms by running DB2START on those members.

DB2 online cloning with removal of data sharing members procedure

Refer to these instructions if an online clone of a data sharing group with removal of members is desired. An online cloning occurs when DB2 SET LOG SUSPEND is issued to suspend transactions, or when a consistent copy process is used such as consistent FlashCopy, SnapShot, or TimeFinder/Clone, or consistent split or break mirror.

Important: If utilities might be running or registered in SYSUTILX on the source DB2 subsystem when it is cloned, SYSUTILX and its indexes should be cleaned out when DB2 conditioning is complete. You can use the DB2UTILXCLEAN command for this purpose. If SYSUTILX is not cleaned out, the source DB2 subsystem might become corrupted when the target DB2 subsystem issues a **-TERM UTIL** command to clean up the entries it finds in SYSUTILX.

Important: To prevent source corruption, ensure that for the special zparm DSNZSPEC, the macro DSN6SPRM was changed to use DEFER,ALL. Refer to the topic "Cloning a DB2 subsystem" on page 80 for information on how to set up the special dsnzparm module.

This procedure requires the following:

- All members of the group should also be cloned.
- The target DB2 XCF structures should be deallocated prior to the first starting of the target DB2 subsystem.
- The new target DB2 data sharing group members will have no log history, so image copies must be taken if further recovery is desired.
- The special zparms will need to be set up for each target member. For more information, see the topic "Cloning a DB2 subsystem" on page 80.
- Because the source DB2 subsystems are active at the time of the clone, any one of them may have a lock outstanding on a DB2 table. These outstanding locks are carried into the target DB2 sharing environment. To free these locks in the target environment, the corresponding target member will need to be started in maintenance mode with its own special zparms.
- A target DB2 subsystem will use the same buffer pool specifications as its corresponding source DB2 subsystem. If the buffer pool definitions in the source

DB2 subsystem are large, care should be taken that sufficient real and auxiliary storage exists to support the size of the buffer pools in the target DB2 subsystem until ALTER BUFFERPOOL commands can be issued.

- The names of the work databases are not changed as part of the cloning. The target DB2 system will use the same database names as the source DB2 system. For data sharing, if you want the work database names in the target DB2 system to include a target member identifier, the work databases will need to be manually dropped and created with the desired names.

The following table provides the steps for online cloning with removal of data sharing members:

Table 22. DB2 online cloning with removal of data sharing members procedure

Step	Description
1	Run the DB2SETLOG command, SCKZJCL member (CKZDSETL). This step is not necessary if using consistent FlashCopy, SnapShot, or TimeFinder/Clone, or consistent split or break mirror.
2	Run the COPY command, SCKZJCL member (CKZCOPY).
3	Run the DB2SETLOG command, SCKZJCL member (CKZDSETL). This step is not necessary if using consistent FlashCopy, SnapShot, or TimeFinder/Clone, or consistent split or break mirror.
4	Run the RENAME command, SCKZJCL member (CKZREN).
5	Run the DB2UPDATE command, SCKZJCL member (CKZDUPD).
6	Run the DB2UPDATE command again, SCKZJCL member (CKZDUPD2).
7	Run the DB2START command, SCKZJCL member (CKZDSTA).
8	Run the DB2START command, SCKZJCL member (CKZDSTA).
9	Run the DB2FIX command, SCKZJCL member (CKZDFIX), to correct any of the DB2 catalog (DSNDB06) and DB2 Directory (DSNDB01) page spaces that are restricted.
	Attention: Steps 10, 11, and 12 are required only if the table space DBD01 (SYSDBDXA for DB2 10 NFM) in database DSNDB01 was restricted when DB2FIX ran. The changes made to DBD01 (SYSDBDXA for DB2 10 NFM) by DB2UPDATE may have been regressed and must be redone. Otherwise, proceed from step 9 directly to step 13.
10	(Optional, see note.) Run the DB2STOP command, SCKZJCL member (CKZDSTO).
11	(Optional, see note.) Run the DB2UPDATE command again, SCKZJCL member (CKZDUPD).
12	(Optional, see note.) Run the DB2START command, SCKZJCL member (CKZDSTA).
13	Run the DB2SQL command, SCKZJCL member (CKZDSQL).
14	Run the DB2FIX command, SCKZJCL member (CKZDFIX), to correct any of the application page spaces that are restricted..
15	Run the DB2STOP command, SCKZJCL member (CKZDSTO).
16	Run the DB2RBLDBSDS command, SCKZJCL member (CKZDRBBS).
17	Run the DB2LGRNXCLEAN command, SCKZJCL member (CKZDLGCL).
18	Run the DB2XCFCLEAN command, SCKZJCL member (CKZDXCFC).
19	(Optional) Run the DB2UTILXCLEAN command, SCKZJCL member (CKZDUTCL).
20	Run the DB2START command, SCKZJCL member (CKZDSTA).
21	Set the DB2 buffer pool attributes.

Steps for DB2 online cloning with removal of data sharing members

These are the steps for DB2 online cloning with removal of data sharing members.

Procedure

1. Run the DB2SETLOG command.

Note: This step is not necessary if using consistent FlashCopy, SnapShot, or TimeFinder/Clone, or consistent split or break mirror.

The DB2 Cloning Tool DB2SETLOG command with the SUSPEND keyword issues a SET LOG LOGLOAD(0) command to force any pending database writes to disk, followed by a SET LOG SUSPEND command to suspend update activity and flush the log buffers to disk. On the source system, run the DB2SETLOG command with the SUSPEND keyword, SCKZJCL member (CKZDSETL).

Run the DB2SETLOG command with the SUSPEND keyword, SCKZJCL member (CKZDSETL), for each member in a data sharing environment.

2. Run the COPY command. Copy the source volumes to the target volumes and back up the source ICF catalogs that point to the data sets on the source volumes being copied.

If you are using FlashCopy, SnapShot, or TimeFinder/Clone, run the COPY command using SCKZJCL member (CKZCOPY) to copy the source volumes to the target volumes and back up the source ICF catalogs that point to the data sets on the source volumes being copied. See Chapter 23, "DB2 Cloning Tool Subsystem Cloning commands," on page 337 for more information on the COPY command.

If you are using a 'Split of a Continuous Mirror' tool, this step assumes the mirror relationship has been started in advance of when you want to 'split or suspend' the mirror relationship. Issue the 'split or suspend' then run COPY using SCKZJCL member (CKZCOPY) to back up the source ICF catalogs that point to the data sets on the source volumes being copied.

All members of the DB2 data sharing group should be cloned.

3. Run the DB2SETLOG command.

Note: This step is not necessary if using consistent FlashCopy, SnapShot, or TimeFinder/Clone, or consistent split or break mirror.

After the data has been copied to the target volumes, the source ICF catalogs have been backed up, and COPY has completed successfully, you can resume access to the source volumes.

Run the DB2SETLOG command with the RESUME keyword, SCKZJCL member (CKZDSETL), on the source system to resume update activity. Run the DB2SETLOG command with the RESUME keyword, SCKZJCL member (CKZDSETL), for each member in a data sharing environment.

4. Run the RENAME command. Run the RENAME command using SCKZJCL member (CKZREN) to rename and catalog the target volume data sets.

5. Run the DB2UPDATE command. Run the DB2UPDATE command using SCKZJCL member (CKZDUPD). The source DB2 is data sharing so the DB2-GROUP and DB2-MEMBERS keywords must be used.

The DB2UPDATE command makes the necessary DB2 changes to reflect the renamed data sets. DB2UPDATE updates the DB2 directory and the DB2 boot strap data sets (BSDSs).

- DB2 directory updates – the VCATNAME, and optionally, the DB2 storage group names are updated.
- BSDS updates – the DB2 catalog name, and 'active' log data set names are updated. Optionally, the ARCHIVE data set names and volume serial numbers in the BSDS are updated.

The target DB2 XCF structures need to be deallocated. The DB2-XCFCLEAN(Y) keyword, the default, will be used to instruct DB2UPDATE to deallocate the target DB2 XCF structures. If this keyword is set to (N), the target DB2 XCF structures need to be manually deallocated prior to starting the target DB2 subsystem. See “Deallocating target DB2 coupling facility structures” on page 591 for more information on this process.

6. **Run the DB2UPDATE command again.** Run the DB2UPDATE command again, this time using SCKZJCL member (CKZDUPD2) for each additional data sharing member. This step is in addition to the DB2UPDATE command SCKZJCL member (CKZDUPD) previously executed. The DB2UPDATE command using SCKZJCL member (CKZDUPD2) makes the same changes to each subsequent member's BSDSs as the DB2UPDATE command in the previous step. However, CKZDUPD2 does not do any DB2 directory updates.
7. **Run the DB2START command.**

Important: To prevent source corruption, ensure that for the special zparm DSNZSPEC, the macro DSN6SPRM was changed to use DEFER,ALL.. Refer to the topic “Cloning a DB2 subsystem” on page 80 for information on how to set up the special dsnzparm module.

Run the DB2START command to start the target subsystem in maintenance mode using the special zparms. Use the SPECIAL and DSNZPARM keywords on the target subsystem using SCKZJCL member (CKZDSTA).

Note: The initial starts of the target DB2 subsystems should not be a cold start. A cold start will not allow the target DB2 to resolve in-flight transactions and could leave inconsistencies in DB2 data.

The following error messages are expected. If an in-flight transaction was in progress, multiple messages may be associated with it:

```
DSNI001I @TGT1 RESTART HAS BEEN DEFERRED
          REASON 00C90095
          TYPE 00000200
          NAME databasename .tablespace or indexspace name
DSNB250E @TGT1 DSNIIIMPD A PAGE RANGE WAS ADDED TO THE LOGICAL PAGE LIST
          DATABASE NAME=database name
          SPACE NAME=tablespace or indexspace name
          DATA SET NUMBER=1
          PAGE RANGE X'nnnnnnnn' TO X'nnnnnnnn'
          START LRSN=X'nnnnnnnnnnnn'
          END LRSN=X'nnnnnnnnnnnn'
          START RBA=X'nnnnnnnnnnnn'
```

8. **Run the DB2START command again.**

Important: To prevent source corruption, ensure that for the special zparm DSNZSPEC, the macro DSN6SPRM was changed to use DEFER,ALL.. Refer to the topic “Cloning a DB2 subsystem” on page 80 for information on how to set up the special dsnzparm module.

The other target DB2 members may hold locks that need to be released before the DB2FIX command can be run. To release these locks, start the remaining target DB2 members in maintenance mode with LIGHT(YES) and the special

zparms as described in the next paragraph. LIGHT(YES) causes the target DB2 subsystem to stop after it has released any locks it held.

Run the DB2START command with the SPECIAL, LIGHT, and DSNZPARM keywords on the target subsystem using SCKZJCL member (CKZDSTA).

Note: The initial starts of the target DB2 subsystems should not be a cold start. A cold start will not allow the target DB2 to resolve in-flight transactions and could leave inconsistencies in DB2 data.

9. **Run the DB2FIX command.** Run the DB2FIX command to correct any of the DB2 catalog (DSNDB06) and DB2 Directory (DSNDB01) page spaces that are restricted.

Run the DB2FIX command using SCKZJCL member (CKZDFIX) with the keyword DATABASES(DB2) on the target subsystem. This will start any page spaces in the DB2 catalog (DSNDB06) and DB2 Directory (DSNDB01) that have LPL or GRECP status.

Only one DB2 member should be active when DB2FIX is run.

There may be times when locks are held by other members even though they have been started. The use of MEMBERS-NEED-STARTING(ACTION(CONTINUE)) can be used to cause DB2FIX to issue the START DATABASE commands for table and index spaces in LPL or GRECP status even though other members in the data sharing group have locks outstanding. If a timeout occurs when using MEMBERS-NEED-STARTING(ACTION(CONTINUE)), it may be necessary to run DB2FIX using MEMBERS-NEED-STARTING(ACTION(CONTINUE)) on the other members as well.

Attention: Steps 10, 11, and 12 need to be performed ONLY if the table space DBD01 (SYSDBDXA for DB2 10 NFM) in database DSNDB01 was restricted when DB2FIX ran. The changes made to DBD01 (SYSDBDXA for DB2 10 NFM) by DB2UPDATE may have been regressed and need to be redone. These steps can be automated by using either the DB2FIX return code setting, or the DB2FIX WTO message produced when DB2FIX starts restricted page spaces. For more information, see the "DB2FIX" on page 379 command. Otherwise, proceed to step 13.

10. **Run the DB2STOP command.** Run the DB2STOP command using SCKZJCL member (CKZDSTO) to stop the target DB2 subsystem from running in maintenance mode.
11. **Run the DB2UPDATE command again.** Run the DB2UPDATE command with the DBD01ONLY keyword using SCKZJCL member (CKZDUPD). The DB2UPDATE command makes the necessary DB2 changes to reflect the renamed data sets. DB2UPDATE with the DBD01ONLY keyword updates only the DB2 directory. The VCATNAME, and optionally, the DB2 storage group names are updated.

The target DB2 XCF structures need to be deallocated. The DB2-XCFCLEAN(Y) keyword, the default, will be used to instruct DB2UPDATE to deallocate the target DB2 XCF structures. If this keyword is set to (N), the target DB2 XCF structures need to be manually deallocated prior to starting the target DB2 subsystem. See "Deallocating target DB2 coupling facility structures" on page 591 for more information on this process.

12. **Run the DB2START command.**

Important: To prevent source corruption, ensure that for the special zparm DSNZSPEC, the macro DSN6SPRM was changed to use DEFER,ALL.. Refer to the topic "Cloning a DB2 subsystem" on page 80 for information on how to set up the special dsnzparm module.

Run the DB2START command to start the target DB2 subsystem in maintenance mode using the special zparms. Use the SPECIAL and DSNZPARM keywords on the target subsystem using SCKZJCL member (CKZDSTA).

13. **Run the DB2SQL command.** Run the DB2SQL command using SCKZJCL member (CKZDSQL) to update the DB2 catalog. This step assumes that the plan and package have already been installed. For more information on installing the plan and package, see the topic “Cloning a DB2 subsystem” on page 80.

The DB2SQL command makes the necessary changes to the DB2 catalog. The changes include the VCATNAME, storage group names, and volumes.

14. **Run the DB2FIX command.** Run the DB2FIX command to correct any of the application page spaces that are restricted. Run the DB2FIX command using SCKZJCL member (CKZDFIX) with the keyword DATABASES(APPLICATION) on the target subsystem. This will start any application page spaces that have LPL or GRECP status.

Only one DB2 member should be active when DB2FIX is run.

There may be times when locks are held by other members even though they have been started. The use of MEMBERS-NEED-STARTING(ACTION(CONTINUE)) can be used to cause DB2FIX to issue the START DATABASE commands for table and index spaces in LPL or GRECP status even though other members in the data sharing group have locks outstanding. If a timeout occurs when using MEMBERS-NEED-STARTING(ACTION(CONTINUE)),it may be necessary to run DB2FIX using MEMBERS-NEED-STARTING(ACTION(CONTINUE)) on the other members as well.

15. **Run the DB2STOP command.** To stop the target DB2 subsystem from running in maintenance mode, run the DB2STOP command using SCKZJCL member (CKZDSTO).
16. **Run the DB2RBLDBSDS command.** Run the DB2RBLDBSDS command using SCKZJCL member (CKZDRBBS) for each desired target data sharing member. This step will rebuild the member BSDS to have only active logs in it.
17. **Run the DB2LGRNXCLEAN command.** To clean out all information in SYSLGRNX, run the DB2LGRNXCLEAN command using SCKZJCL member (CKZDLGCL).
18. **Run the DB2XCFCLEAN command.** Run the DB2XCFCLEAN command using SCKZJCL member (CKZDXCFC). This step will deallocate the DB2 XCF structures and remove the XCF group members.
19. **(Optional) Run the DB2UTILXCLEAN command.** To clean out all information in SYSUTILX, run the DB2UTILXCLEAN command using SCKZJCL member (CKZDUTCL). If utilities may have been running or registered in SYSUTILX when the source DB2 subsystem is cloned, SYSUTILX and its indexes should be cleaned out. There might be table and index spaces that have UT status due to utilities that were running or registered in SYSUTILX when the source DB2 subsystem was cloned.
20. **Run the DB2START command.** To start the target DB2 subsystem using its 'normal' zparms, run the DB2START command using SCKZJCL member (CKZDSTA). You can also start other members of the target data sharing group with their normal zparms by running DB2START on those members.
21. **Set the DB2 buffer pool attributes.** The DB2 buffer pool attributes need to be set to the desired values. The DB2 -ALTER BUFFERPOOL command can be used to set desired buffer pool attributes.

DB2 online cloning with target becoming non-data sharing procedure

Refer to these instructions if an online clone of a data sharing group with the target becoming non-data sharing is desired. An online cloning occurs when DB2 SET LOG SUSPEND is issued to suspend transactions, or when a consistent copy process is used such as consistent FlashCopy, SnapShot, or TimeFinder/Clone, or consistent split or break mirror.

Important: If utilities might be running or registered in SYSUTILX on the source DB2 subsystem when it is cloned, SYSUTILX and its indexes should be cleaned out when DB2 conditioning is complete. You can use the DB2UTILXCLEAN command for this purpose. If SYSUTILX is not cleaned out, the source DB2 subsystem might become corrupted when the target DB2 subsystem issues a **-TERM UTIL** command to clean up the entries it finds in SYSUTILX.

Important: To prevent source corruption, ensure that for the special zparm DSNZSPEC, the macro DSN6SPRM was changed to use DEFER,ALL. Refer to the topic “Cloning a DB2 subsystem” on page 80 for information on how to set up the special dsnzparm module.

This procedure requires the following:

- All members of the data sharing group should be cloned.
- The target DB2 XCF structures should be deallocated prior to the first starting of the target DB2 subsystem.
- The initial start-up of the target DB2 system must be done in data sharing mode. This is necessary because this is an online cloning and the resolution of work in flight by the target DB2 must be done in the same mode as the source DB2.
- The special zparks will need to be set up for each target member. For more information, see the topic “Cloning a DB2 subsystem” on page 80.
- Because the source DB2 subsystems are active at the time of the clone, any one of them may have a lock outstanding on a DB2 table. These outstanding locks are carried into the target DB2 sharing environment. To free these locks in the target environment, the corresponding target member will need to be started in maintenance mode with its own special zparks.
- The special zparks used by the DB2 subsystems need to be data sharing and the normal zparks used by the final target DB2 subsystems needs to be non-data sharing.
- A target DB2 subsystem will use the same buffer pool specifications as its corresponding source DB2 subsystem. If the buffer pool definitions in the source DB2 subsystem are large, care should be taken that sufficient real and auxiliary storage exists to support the size of the buffer pools in the target DB2 subsystem until ALTER BUFFERPOOL commands can be issued.
- The names of the work databases are not changed as part of the cloning. The target DB2 system will use the same database names as the source DB2 system. For data sharing, if you want the work database names in the target DB2 system to include a target member identifier, the work databases will need to be manually dropped and created with the desired names.

The following table provides the steps for online cloning with the target becoming a non-data sharing subsystem:

Table 23. DB2 online cloning with target becoming non-data sharing procedure

Step	Description
1	Run the DB2SETLOG command, SCKZJCL member (CKZDSETL). This step is not necessary if using consistent FlashCopy, SnapShot, or TimeFinder/Clone, or consistent split or break mirror.
2	Run the COPY command, SCKZJCL member (CKZCOPY).
3	Run the DB2SETLOG command, SCKZJCL member (CKZDSETL). This step is not necessary if using consistent FlashCopy, SnapShot, or TimeFinder/Clone, or consistent split or break mirror.
4	Run the RENAME command, SCKZJCL member (CKZREN).
5	Run the DB2UPDATE command, SCKZJCL member (CKZDUPD).
6	Run the DB2UPDATE command again, SCKZJCL member (CKZDUPD2).
7	Run the DB2START command, SCKZJCL member (CKZDSTA).
8	Run the DB2START command, SCKZJCL member (CKZDSTA), to start the remaining target DB2 members.
9	Run the DB2FIX command, SCKZJCL member (CKZDFIX).
	Attention: Steps 10, 11, and 12 are required only if the table space DBD01 (SYSDBDXA for DB2 10 NFM) in database DSNDB01 was restricted when DB2FIX ran. The changes made to DBD01 (SYSDBDXA for DB2 10 NFM) by DB2UPDATE may have been regressed and must be redone. Otherwise, proceed from step 9 directly to step 13.
10	(Optional, see note.) Run the DB2STOP command, SCKZJCL member (CKZDSTO).
11	(Optional, see note.) Run the DB2UPDATE command again, SCKZJCL member (CKZDUPD).
12	(Optional, see note.) Run the DB2START command, SCKZJCL member (CKZDSTA).
13	Run the DB2SQL command, SCKZJCL member (CKZDSQL).
14	Run the DB2FIX command, SCKZJCL member (CKZDFIX), to correct any of the application page spaces that are restricted.
15	Run the DB2STOP command, SCKZJCL member (CKZDSTO).
16	Run the DB2RBLDBSDS command, SCKZJCL member (CKZDRBBS).
17	Run the DB2XCFCLEAN command, SCKZJCL member (CKZDXCFC).
18	(Optional) Run the DB2UTILXCLEAN command, SCKZJCL member (CKZDUTCL).
19	Run the DB2START command, SCKZJCL member (CKZDSTA).

Steps for DB2 online cloning with target becoming non-data sharing

These are the steps for DB2 online cloning with target becoming non-data sharing.

Procedure

1. Run the DB2SETLOG command.

Note: This step is not necessary if using consistent FlashCopy, SnapShot, or TimeFinder/Clone, or consistent split or break mirror.

The DB2 Cloning Tool DB2SETLOG command with the SUSPEND keyword issues a SET LOG LOGLOAD(0) command to force any pending database writes to disk, followed by a SET LOG SUSPEND command to suspend

update activity and flush the log buffers to disk. On the source system, run the DB2SETLOG command with the SUSPEND keyword, SCKZJCL member (CKZDSETL).

Run the DB2SETLOG command with the SUSPEND keyword, SCKZJCL member (CKZDSETL), for each member in a data sharing environment.

2. **Run the COPY command.** Copy the source volumes to the target volumes and back up the source ICF catalogs that point to the data sets on the source volumes being copied.

If you are using FlashCopy, SnapShot, or TimeFinder/Clone, run the COPY command using SCKZJCL member (CKZCOPY) to copy the source volumes to the target volumes and back up the source ICF catalogs that point to the data sets on the source volumes being copied. See Chapter 23, “DB2 Cloning Tool Subsystem Cloning commands,” on page 337 for more information on the COPY command.

If you are using a 'Split of a Continuous Mirror' tool, this step assumes the mirror relationship has been started in advance of when you want to 'split or suspend' the mirror relationship. Issue the 'split or suspend' then run COPY using SCKZJCL member (CKZCOPY) to back up the source ICF catalogs that point to the data sets on the source volumes being copied.

All members of the DB2 data sharing group should be cloned.

3. **Run the DB2SETLOG command.**

Note: This step is not necessary if using consistent FlashCopy, SnapShot, or TimeFinder/Clone, or consistent split or break mirror.

After the data has been copied to the target volumes, the source ICF catalogs have been backed up, and COPY has completed successfully, you can resume access to the source volumes.

Run the DB2SETLOG command with the RESUME keyword, SCKZJCL member (CKZDSETL), on the source system to resume update activity. Run the DB2SETLOG command with the RESUME keyword, SCKZJCL member (CKZDSETL), for each member in a data sharing environment.

4. **Run the RENAME command.** Run the RENAME command using SCKZJCL member (CKZREN) to rename and catalog the target volume data sets.
5. **Run the DB2UPDATE command.** Run the DB2UPDATE command using SCKZJCL member (CKZDUPD). The source DB2 is data sharing so the DB2-GROUP and DB2-MEMBERS keywords must be used.

The DB2UPDATE command makes the necessary DB2 changes to reflect the renamed data sets. DB2UPDATE updates the DB2 directory and the DB2 boot strap data sets (BSDSs).

- DB2 directory updates – the VCATNAME, and optionally, the DB2 storage group names are updated.
- BSDS updates – the DB2 catalog name, and 'active' log data set names are updated. Optionally, the ARCHIVE data set names and volume serial numbers in the BSDS are updated.

The target DB2 XCF structures need to be deallocated. The DB2-XCFCLEAN(Y) keyword, the default, will be used to instruct DB2UPDATE to deallocate the target DB2 XCF structures. If this keyword is set to (N), the target DB2 XCF structures need to be manually deallocated prior to starting the target DB2 subsystem. See “Deallocating target DB2 coupling facility structures” on page 591 for more information on this process.

6. **Run the DB2UPDATE command again.** Run the DB2UPDATE command again, this time using SCKZJCL member (CKZDUPD2) for each additional

data sharing member. This step is in addition to the DB2UPDATE command SCKZJCL member (CKZDUPD) previously executed. The DB2UPDATE command using SCKZJCL member (CKZDUPD2) makes the same changes to each subsequent member's BSDs as the DB2UPDATE command in the previous step. However, CKZDUPD2 does not do any DB2 directory updates. The source DB2 is data sharing so the DB2-GROUP and DB2-MEMBERS keywords must be used.

7. Run the DB2START command.

Important: To prevent source corruption, ensure that for the special zparm DSNZSPEC, the macro DSN6SPRM was changed to use DEFER,ALL. Refer to “Cloning a DB2 subsystem” on page 80 for information on how to set up the special dsnzparm module.

Run the DB2START command to start the target subsystem in maintenance mode using the special zparms. Use the SPECIAL and DSNZPARM keywords on the target subsystem using SCKZJCL member (CKZDSTA).

This start-up of the target DB2 subsystem must be done in data sharing mode. This is necessary because this is an online cloning and the resolution of work in flight by the target DB2 must be done in the same mode as the source DB2.

Note: The initial starts of the target DB2 subsystems should not be a cold start. A cold start will not allow the target DB2 to resolve in-flight transactions and could leave inconsistencies in DB2 data.

The following error messages are expected. If an in-flight transaction was in progress, multiple messages may be associated with it:

```
DSNI001I @TGT1 RESTART HAS BEEN DEFERRED
          REASON 00C90095
          TYPE 00000200
          NAME databasename .tablespace or indexspace name
DSNB250E @TGT1 DSNIIIMP A PAGE RANGE WAS ADDED TO THE LOGICAL PAGE LIST
          DATABASE NAME=database name
          SPACE NAME=tablespace or indexspace name
          DATA SET NUMBER=1
          PAGE RANGE X'nnnnnnnn' TO X'nnnnnnnn'
          START LRSN=X'nnnnnnnnnnnn'
          END LRSN=X'nnnnnnnnnnnn'
          START RBA=X'nnnnnnnnnnnn'
```

8. Run the DB2START command again.

Important: To prevent source corruption, ensure that for the special zparm DSNZSPEC, the macro DSN6SPRM was changed to use DEFER,ALL. Refer to the topic “Cloning a DB2 subsystem” on page 80 for information on how to set up the special dsnzparm module.

The other target DB2 members may hold locks that need to be released before the DB2FIX command can be run. To release these locks, start the remaining target DB2 members in maintenance mode with LIGHT(YES) and the special zparms as described in the next paragraph. LIGHT(YES) causes the target DB2 subsystem to stop after it has released any locks it held.

Run the DB2START command with the SPECIAL, LIGHT, and DSNZPARM keywords on the target subsystem using SCKZJCL member (CKZDSTA).

This start-up of the target DB2 subsystems must be done in data sharing mode. This is necessary because this is an online cloning and the resolution of work in flight by the target DB2 must be done in the same mode as the source DB2.

Note: The initial starts of the target DB2 subsystems should not be a cold start. A cold start will not allow the target DB2 to resolve in-flight transactions and could leave inconsistencies in DB2 data.

9. **Run the DB2FIX command.** Run the DB2FIX command to correct any of the DB2 catalog (DSNDB06) and DB2 Directory (DSNDB01) page spaces that are restricted.

Run the DB2FIX command using SCKZJCL member (CKZDFIX) with the keyword DATABASES(DB2) on the target subsystem. This will start any page spaces in the DB2 catalog (DSNDB06) and DB2 Directory (DSNDB01) that have LPL or GRECP status.

Only one DB2 member should be active when DB2FIX is run.

There may be times when locks are held by other members even though they have been started. The use of MEMBERS-NEED-STARTING(ACTION(CONTINUE)) can be used to cause DB2FIX to issue the START DATABASE commands for table and index spaces in LPL or GRECP status even though other members in the data sharing group have locks outstanding. If a timeout occurs when using MEMBERS-NEED-STARTING(ACTION(CONTINUE)), it may be necessary to run DB2FIX using MEMBERS-NEED-STARTING(ACTION(CONTINUE)) on the other members as well.

Attention: Steps 10, 11, and 12 need to be performed ONLY if the table space DBD01 (SYSDBDXA for DB2 10 NFM) in database DSNDB01 was restricted when DB2FIX ran. The changes made to DBD01 (SYSDBDXA for DB2 10 NFM) by DB2UPDATE may have been regressed and need to be redone. These steps can be automated by using either the DB2FIX return code setting, or the DB2FIX WTO message produced when DB2FIX starts restricted page spaces. For more information, see the DB2FIX command in Chapter 23, “DB2 Cloning Tool Subsystem Cloning commands,” on page 337. Otherwise, proceed to step 13.

10. **Run the DB2STOP command.** Run the DB2STOP command using SCKZJCL member (CKZDSTO) to stop the target DB2 subsystem from running in maintenance mode.
11. **Run the DB2UPDATE command again.** Run the DB2UPDATE command with the DBD01ONLY keyword using SCKZJCL member (CKZDUPD). The DB2UPDATE command makes the necessary DB2 changes to reflect the renamed data sets. DB2UPDATE with the DBD01ONLY keyword updates only the DB2 directory. The VCATNAME, and optionally, the DB2 storage group names are updated.

The target DB2 XCF structures need to be deallocated. The DB2-XCFCLEAN(Y) keyword, the default, will be used to instruct DB2UPDATE to deallocate the target DB2 XCF structures. If this keyword is set to (N), the target DB2 XCF structures need to be manually deallocated prior to starting the target DB2 subsystem. See “Deallocating target DB2 coupling facility structures” on page 591 for more information on this process.

12. **Run the DB2START command.**

Important: To prevent source corruption, ensure that for the special zparm DSNZSPEC, the macro DSN6SPRM was changed to use DEFER,ALL. Refer to the topic “Cloning a DB2 subsystem” on page 80 for information on how to set up the special dsnzparm module.

Run the DB2START command to start the target DB2 subsystem in maintenance mode using the special zparms. Use the SPECIAL and DSNZPARM keywords on the target subsystem using SCKZJCL member (CKZDSTA).

This start-up of the target DB2 subsystem must be done in data sharing mode.

13. **Run the DB2SQL command.** Run the DB2SQL command using SCKZJCL member (CKZDSQL) to update the DB2 catalog. This step assumes that the plan and package have already been installed. For more information on installing the plan and package, see the topic “Cloning a DB2 subsystem” on page 80.

The DB2SQL command makes the necessary changes to the DB2 catalog. The changes include the VCATNAME, storage group names, and volumes.

14. **Run the DB2FIX command.** Run the DB2FIX command to correct any of the application page spaces that are restricted. Run the DB2FIX command using SCKZJCL member (CKZDFIX) with the keyword DATABASES(APPLICATION) on the target subsystem. This will start any application page spaces that have LPL or GRECP status.

Only one DB2 member should be active when DB2FIX is run.

There may be times when locks are held by other members even though they have been started. The use of MEMBERS-NEED-STARTING(ACTION(CONTINUE)) can be used to cause DB2FIX to issue the START DATABASE commands for table and index spaces in LPL or GRECP status even though other members in the data sharing group have locks outstanding. If a timeout occurs when using MEMBERS-NEED-STARTING(ACTION(CONTINUE)), it may be necessary to run DB2FIX using MEMBERS-NEED-STARTING(ACTION(CONTINUE)) on the other members as well.

15. **Run the DB2STOP command.** To stop the target DB2 subsystem from running in maintenance mode, run the DB2STOP command using SCKZJCL member (CKZDSTO).
16. **Run the DB2RBLDBSDS command.** To alter the member BSDS to be non-data sharing, run the DB2RBLDBSDS command using SCKZJCL member (CKZDRBBS) for the desired target member using the keyword DATA-SHARING(NO).
17. **Run the DB2XCFCLEAN command.** Run the DB2XCFCLEAN command using SCKZJCL member (CKZDXCFC). This step will deallocate the DB2 XCF structures and remove the XCF group members.

18. **(Optional) Run the DB2UTILXCLEAN command.** To clean out all information in SYSUTILX, run the DB2UTILXCLEAN command using SCKZJCL member (CKZDUTCL). If utilities may have been running or registered in SYSUTILX when the source DB2 subsystem is cloned, SYSUTILX and its indexes should be cleaned out. There might be table and index spaces that have UT status due to utilities that were running or registered in SYSUTILX when the source DB2 subsystem was cloned.

19. **Run the DB2START command.** To start the target DB2 subsystem with its normal (non-data sharing) zparms, run the DB2START command using the REPLY-TO-RESTART-WTOR(Y) keyword on the target subsystem using SCKZJCL member (CKZDSTA). This start-up of the target DB2 subsystem will require a cold start, so an operator reply will need to be made to the DB2 message:

```
DSNJ246I  CONDITIONAL RESTART RECORD INDICATES COLD START AT  
RBA xxxxxxxxxxxx.  REPLY Y TO CONTINUE, N TO CANCEL
```

Using the REPLY-TO-RESTART-WTOR(Y) keyword with the DB2START command will cause DB2START to automatically reply to the DSNJ246I WTOR message.

Chapter 10. Subsystem cloning using the DB2 Cloning Tool stored procedure

DB2 Cloning Tool provides a DB2 stored procedure that can generate the necessary jobs to perform the subsystem cloning, schedule the jobs in the DB2 administrative task scheduler, and monitor the execution of the jobs. The stored procedure will return to the caller when the requested cloning has ended, either in success or failure.

The cloning may be from volumes or may be from a DB2 system level backup created by the DB2 BACKUP SYSTEM command. For data sharing, the target DB2 system may optionally have fewer members or be non-data sharing. The cloning may include one or more non data sharing subsystems or data sharing groups. The cloned DB2 systems must reside on the set of source volumes being used.

About the DB2 Cloning Tool stored procedure

This topic describes the stored procedure inputs and outputs and system requirements for using the stored procedure.

Stored procedure input and output

The input to the stored procedure is:

- The requested operation/type of the run
- The DSN and optional member name of the product parameter file
- The DSN and optional member name of the DB2 systems parameter file
- The DSN and optional member name of the cloning parameter file

The **product parameter file** provides necessary information about the product. This information includes the DSN of the product load library and the DSN of the CKZINI file.

The **DB2 systems parameter file** provides necessary information about the DB2 systems that will be used by the cloning. This information includes items for each DB2 system, such as the SSID, DSN of the SDSNLOAD library, the system that the DB2 should run on, the system VCAT, and DSNs of the boot strap data sets.

The **cloning parameter file** provides necessary information about the specific cloning. This information includes the job cards, userid and password the jobs will run under, the DSN to build the JCL into, the prefix to use for any work file data sets, the source and target volumes, the source and target ICF catalogs, the rename masks, and the source and target DB2 systems.

The output from the stored procedure is:

- A return code indicating the success or failure of the requested operation
- A message area containing messages about the success or failure of the requested operation

System requirements

The following requirements must be met to use the DB2 Cloning Tool Subsystem Cloning stored procedure:

- The DB2 administrative task scheduler must be configured and available on the DB2 system where the stored procedure is run.
- Only the DB2 Version 9.1 and DB2 10 administrative task schedulers are supported. For DB2 Version 9.1, APAR/PTF PM02658/UK60388 must be applied. The DB2 Version 8 administrative task scheduler is not supported.
- If the WLM address space the stored procedure runs in does not include the DB2 Cloning Tool load library, the following modules must be copied from the DB2 Cloning Tool load library into a load library that is available to the WLM address space:
 - CKZ00200
 - CKZ00201
 - CKZ00900
 - CKZ01DYN
 - CKZ01HEX
 - CKZ01PSN
- If the cloning will suspend or stop the source DB2 system, the source DB2 system should not be the same DB2 system that the stored procedure and the DB2 administrative task scheduler are running on. Suspending the DB2 system causes the DB2 administrative task scheduler to freeze, thus preventing the scheduling of the cloning jobs that will resume the source DB2 system. Stopping a DB2 system causes the DB2 administrative task scheduler to stop, thus preventing the scheduling of the cloning jobs that will start the source DB2 system.

Steps for cloning a DB2 subsystem using the stored procedure

The procedure in this section describes the general steps to clone DB2 subsystems using the stored procedure.

Procedure

1. Set up the product parameter file.
2. Set up the DB2 systems parameter file.
3. Set up the cloning parameter file.
4. Invoke the stored procedure.
5. Verify the cloning.
6. Run the verified cloning.

Parameter files and parameter descriptions

Set up the parameter files that are required for the stored procedure according to the following requirements.

You must create three parameter files that are used by the stored procedure:

- Product parameter file (PPARM)
- DB2 systems parameter file (SPARM)
- Cloning parameter file (CPARM)

Requirements for all parameter files

The three parameter input files must be defined as RECFM=FB, LRECL=80. Each parameter file can be in a sequential data set or can be a member of a PDS or PDSE.

An asterisk (*) in column 1 denotes a comment. Any other non-blank character in column 1 denotes the start of a parameter and is of the format **keyword = value**. A value can consist of multiple items that are separated by one or more blanks. A value can be continued by using a blank in column 1 of the following record. The value that starts in column 2 is concatenated to the prior records column 72.

PPARM parameter file keywords

The product parameter file provides required information about DB2 Cloning Tool. A sample parameter file can be found in SCKZPPARM member CKZPPARM. Possible product parameter keywords are:

CKZINI

(Required) The DSN and member that identifies the CKZINI to be used.

SCKZLOAD

(Required) The DSN of the DB2 Cloning Tool load library to be used. Multiple load libraries can be specified.

SCKZPARM

The DSN of the DB2 Cloning Tool parm library to be used. This library contains the CKZRNTGT REXX EXEC. This parameter is required if SOURCE-VOLUMES=DB2SLB is specified in the CPARMs.

SCKZDBRM

The DSN of the DB2 Cloning Tool DBRM library to be used. This parameter is required if BIND-ON-TARGET=Y is specified.

SPARM parameter file keywords

The DB2 systems parameter (SPARM) file provides required information about the DB2 system that will be used by the cloning. A sample parameter file can be found in SCKZparm member CKZSPARM. Possible DB2 system parameter keywords follow. All parameters are optional unless otherwise specified.

SSID

(Required) The SSID of the DB2 subsystem. The SSID keyword denotes the start of each DB2 subsystem entry. The first keyword in the DB2 systems parameter file must be SSID.

SDSNLOAD

(Required) The DSN of the DB2 load library to be used. Multiple load libraries can be specified.

SDSNEXIT

The DSN of the DB2 SDSNEXIT load library to be used. Multiple load libraries can be specified.

BSDS01

(Required for the target DB2) The DSN of the BSDS01 data set to be used.

BSDS02

(Required for the target DB2) The DSN of the BSDS02 data set to be used.

| **SPECIAL-DSNZPARM**

| (Required for the target DB2) The name of the special dsnzparm to be used.

| **NORMAL-DSNZPARM**

| The name of the dsnzparm to be used for normal starts of the subsystem.

| **NONDS-DSNZPARM**

| The name of the dsnzparm to be used when starting the surviving member in
| non-data sharing mode.

| **EXEC-SYSTEM**

| The name of the system this DB2 subsystem runs on.

| **GROUP**

| The name of the DB2 group this subsystem belongs to. This parameter is
| required for a data sharing DB2 system.

| **MEMBER**

| The member name of this subsystem. This parameter is required for a data
| sharing DB2 system.

| **SYSVCAT**

| (Required for the target DB2) The system VCAT of this subsystem.

| **DDF-ALIAS**

| The DDF aliases of this subsystem. This parameter corresponds to the ALIAS
| keyword of the DB2UPDATE command.

| **DDF-GENERIC**

| The DDF generic luname of this subsystem. This parameter corresponds to the
| GENERIC keyword of the DB2UPDATE command.

| **DDF-GRPIP4**

| The DDF group IP V4 address for this subsystem. This parameter corresponds
| to the GRPIP4 keyword of the DB2UPDATE command.

| **DDF-GRPIP6**

| The DDF group IP V6 address for this subsystem. This parameter corresponds
| to the GRPIP6 keyword of the DB2UPDATE command.

| **DDF-IPNAME**

| The DDF ipname for this subsystem. This parameter corresponds to the
| IPNAME keyword of the DB2UPDATE command.

| **DDF-IPV4**

| The DDF IP V4 address for this subsystem. This parameter corresponds to the
| IPV4 keyword of the DB2UPDATE command.

| **DDF-IPV6**

| The DDF IP V6 address for this subsystem. This parameter corresponds to the
| IPV6 keyword of the DB2UPDATE command.

| **DDF-LOCATION**

| The DDF location of this subsystem. This parameter corresponds to the
| LOCATION keyword of the DB2UPDATE command.

| **DDF-LUNAME**

| The DDF luname of this subsystem. This parameter corresponds to the
| LUNAME keyword of the DB2UPDATE command.

| **DDF-PASSWORD**

| The DDF password of this subsystem. This parameter corresponds to the
| PASSWORD keyword of the DB2UPDATE command.

DDF-PORT

The DDF port of this subsystem. This parameter corresponds to the PORT keyword of the DB2UPDATE command.

DDF-RESPORT

The DDF resport of this subsystem. This parameter corresponds to the RESPORT keyword of the DB2UPDATE command.

DDF-SECPORT

The DDF secpport of this subsystem. This parameter corresponds to the SECPORT keyword of the DB2UPDATE command.

CPARM parameter file keywords

The cloning parameter (CPARM) file provides required information about the specific cloning. A sample parameter file can be found in SCKZPARM member CKZCPARM. Possible cloning parameter keywords follow. All parameters are optional unless otherwise specified.

CPARM general parameters

JCL-DSN

(Required) The data set name of the library that will contain the JCL for this cloning. This data set must have attributes of DSORG=PO,RECFM=FB,LRECL=80.

STATUS-DSN

(Required) The data set name of the data set that will be used to hold the status of the cloning. This data set must have attributes of DSORG=PS,RECFM=FB,LRECL=200.

TASK-PREFIX

(Required) The prefix that will be used for the task names in the DB2 administrative task scheduler. The maximum length is 16 characters.

USERID

(Required) The userid to be used by the cloning jobs.

PASSWORD

(Required) The password of the userid to be used by the cloning jobs.

WORK-PREFIX

(Required) The prefix that will be used for the data sets created and used by the cloning jobs. The maximum length is 27 characters.

JOBCARD1 - 9

The job cards that will be used by the cloning jobs. JOBCARD1 is required.

CLONING-TYPE

(Required) Specifies the type of cloning. Valid values are OFFLINE or ONLINE.

BIND-ON-TARGET

Indicates whether a bind of the plan and package should be done on the target DB2 before running DB2SQL. Valid values YES | Y or NO | N. This option addresses the situation where the source DB2 system does not have a current bind of the DB2SQL package; therefore, a bind must be run on the target DB2 before running the DB2SQL command.

BIND-PACKAGE

Specifies the package name to use in the bind of the plan and package. This parameter applies only when BIND-ON-TARGET is specified. If not specified, a package name of CKZPACK is used.

CPARM COPY command parameters

These parameters are DB2 Cloning Tool Subsystem Cloning COPY or RESTORE-FROM-DUMPTAPES command parameters. More information about these parameters is available in the topic Chapter 23, "DB2 Cloning Tool Subsystem Cloning commands," on page 337.

FROM-STORAGEGROUP

The source storage groups of the volume copy. This parameter is required if FROM-VOLSER and SOURCE-VOLUMES are not specified. This parameter is mutually exclusive with SOURCE-VOLUMES. This parameter corresponds to the FROM-STORAGEGROUP keyword of the COPY command.

TO-STORAGEGROUP

The target storage groups of the volume copy or restore from system-level backup dump tapes. This parameter is required if TO-VOLSER is not specified. This parameter corresponds to the TO-STORAGEGROUP keyword of the COPY or the RESTORE-FROM-DUMPTAPES command.

FROM-VOLSER

The source volumes of the volume copy. This parameter is required if FROM-STORAGEGROUP and SOURCE-VOLUMES are not specified. This parameter is mutually exclusive with SOURCE-VOLUMES.) This parameter corresponds to the FROM-VOLSER keyword of the COPY command.

TO-VOLSER

The target volumes of the volume copy or restore from system-level backup dump tapes. This parameter is required if TO-STORAGEGROUP is not specified. This parameter corresponds to the TO-VOLSER keyword of the COPY or the RESTORE-FROM-DUMPTAPES command.

EXCLUDE-FROM-VOLSER

The volumes, or volumes that match a mask, to be excluded from being specified for either the FROM-VOLSER parameter or the FROM-STORAGEGROUP parameter. This parameter corresponds to the EXCLUDE-FROM-VOLSER keyword of the COPY command.

EXCLUDE-TO-VOLSER

The volumes, or volumes that match a mask, to be excluded (not selected as targets) from being specified for either the TO-VOLSER parameter or the TO-STORAGEGROUP parameter. This parameter corresponds to the EXCLUDE-TO-VOLSER keyword of the COPY or the RESTORE-FROM-DUMPTAPES command.

SOURCE-BACKINFO-DSN

Specifies the data set name to use for the BACKINFO file. This parameter is used for a cross-sysplex system-level backup cloning. The BACKINFO data set is created by a DB2GETBACKINFO run on the source system and transferred to the target system, where it will be used by the cloning. The DB2GETBACKINFO command is not executed on the target system as part of the cloning. This parameter is used only if SOURCE-VOLUMES= DB2SLB is specified.

SOURCE-DUMP-CLASS

Specifies the dump class of the system-level backup dump tapes to be used for

| the cloning. This parameter is used only if SOURCE-USE-DUMPTAPES = Y or
| YES is specified. This parameter corresponds to the DUMP-CLASS keyword of
| the DB2GETBACKINFO command.

| **SOURCE-USE-DUMPTAPES**

| Specifies that system-level backup dump tapes are used for the cloning. This
| parameter is used only if SOURCE-VOLUMES= DB2SLB is specified. Valid
| values are YES | Y and NO | N.

| **SOURCE-VOLUMES**

| Specifies that the cloning source volumes are from a system-level backup. Valid
| value for this parameter is DB2SLB (the source of the cloning is a DB2 SLB
| created by a DB2 BACKUP SYSTEM). This parameter is mutually exclusive
| with FROM-STORAGEGROUP and FROM-VOLSER.

| **SOURCE-STORAGEGROUP**

| Specifies the source storage group names to be used when pairing the backup
| volumes to the target volumes. This parameter is optional, and will only be
| used if SOURCE-VOLUMES=DB2SLB is specified. The values that are specified
| for this parameter are used for building a FROM-USER-STORAGEGROUP
| keyword in a COPY command or a SOURCE-STORAGEGROUP keyword in a
| RESTORE-FROM-DUMPTAPES command.

| **SOURCE-LOCATION**

| Specifies the location to use to determine the cloning source volumes from a
| system-level backup (SLB). This parameter is required if SOURCE-
| VOLUMES=DB2SLB is specified. This parameter corresponds to the
| LOCATION keyword of the DB2GETBACKINFO command.

| **SOURCE-TOKEN**

| Specifies the token of the backup to be used. A value of LAST uses the latest
| (most recent) backup. This parameter is required if SOURCE-
| VOLUMES=DB2SLB is specified. This parameter corresponds to the TOKEN
| and LAST keywords of the DB2GETBACKINFO command.

| **USERCATALOGS**

| (Required) Specifies the source ICF catalogs that data sets from source (from)
| volumes are cataloged in, and the corresponding target catalog that renamed
| volume data sets are to be cataloged in. This parameter corresponds to the
| USERCATALOGS keyword of the COPY or the RESTORE-FROM-DUMPTAPES
| command.

| **DM-PGM**

| Specifies the data mover program to be used for volume copies. Valid values
| are ADRDSSU or EMCSNAP. This parameter corresponds to the PGM
| subkeyword of the DATA-MOVER keyword of the COPY command. If
| DM-PGM is not specified, ADRDSSU is used for volume copies.

| **DM-BACKGROUND COPY**

| Indicates whether background copy should be done for the volume copies.
| This parameter applies only when DM-PGM=EMCSNAP is specified. This
| parameter corresponds to the BACKGROUND COPY subkeyword of the
| DATA-MOVER keyword of the COPY command.

| **DM-CHECKONLINEPATHSTATUS**

| Indicates that before performing a volume snap, a check will be conducted to
| ensure paths from other CPUs to the target devices are offline. This parameter
| applies only when DM-PGM=EMCSNAP is specified. This parameter
| corresponds to the CHECKONLINEPATHSTATUS subkeyword of the
| DATA-MOVER keyword of the COPY command.

DM-CHECKVTOC

Specifies whether a VTOC analysis of the source volume is performed during copy processing. Valid values are YES | Y (generate the DSS CHECKVTOC keyword) or NO | N (do not generate the DSS CHECKVTOC keyword). This parameter applies only when DM-PGM is not specified or DM-PGM=ADRDUSSU is specified. This parameter corresponds to the CHECKVTOC subkeyword of the DATA-MOVER keyword of the COPY command.

DM-CONSISTENT

Indicates one of the following:

- If DM-PGM is not specified or DM-PGM=ADRDUSSU is specified, indicates whether to use a FlashCopy Consistency Group. Valid values are YES | Y and NO | N.
- If DM-PGM=EMCSNAP is specified, indicates whether to use Enginuity Consistency Assist (ECA) for consistent SNAP VOLUME operations. Valid values are YES | Y and NO | N.

This parameter corresponds to the CONSISTENT subkeyword of the DATA-MOVER keyword of the COPY command.

DM-COPYCMDLIMIT

Specifies the maximum COPY FULL commands that are built by DB2 Cloning Tool for each DSS execution. This parameter applies only when DM-PGM is not specified or DM-PGM=ADRDUSSU is specified. This parameter corresponds to the COPYCMDLIMIT subkeyword of the DATA-MOVER keyword of the COPY command.

DM-DIFFERENTIAL

Indicates whether to use the Enginuity Differential Snap feature for SNAP VOLUME operations. This parameter applies only when DM-PGM=EMCSNAP is specified.

DM-FASTREP

Indicates whether fast replication is preferred (PREF), required (REQ), or not required (NONE). This parameter applies only when DM-PGM is not specified or DM-PGM=ADRDUSSU is specified. This parameter corresponds to the FASTREP subkeyword of the DATA-MOVER keyword of the COPY command.

DM-FCNOCOPY

Indicates whether background copy should be done for the volume copies. Valid values are YES | Y (do not use background copy) or NO | N (use background copy). This parameter applies only when DM-PGM is not specified or DM-PGM=ADRDUSSU is specified. This parameter corresponds to the FCNOCOPY subkeyword of the DATA-MOVER keyword of the COPY command.

DM-FCSETGTOK

Indicates whether a FlashCopy target volume can also be a space efficient volume. Valid values are YES | Y (the target volumes can be space efficient volumes; the FCSETGTOK keyword will be generated) or NO | N (the target volumes cannot be space efficient volumes; the FCSETGTOK keyword will not be generated). This parameter applies only when DM-PGM is not specified or DM-PGM=ADRDUSSU is specified. This parameter corresponds to the FCSETGTOK subkeyword of the DATA-MOVER keyword of the COPY command.

DM-FCTOPPRCPRI

Indicates whether a FlashCopy target volume can also be a PPRC primary volume. Valid values are:

- YES | Y: The target volumes can be PPRC primary volumes. The FCTOPPRCPRI keyword will be generated
- NO | N: The target volumes cannot be PPRC primary volumes. The FCTOPPRCPRI keyword will not be generated.
- PRESMIRREQ | PMR: The FCTOPPRCPRI(PRESMIRREQ) keyword will be generated.
- PRESMIRPREF | PMR: The FCTOPPRCPRI(PRESMIRPREF) keyword will be generated.
- PRESMIRNONE | PMN: The FCTOPPRCPRI(PRESMIRNONE) keyword will be generated.

This parameter applies only when DM-PGM is not specified or DM-PGM=ADRDSSU is specified. This parameter corresponds to the FCTOPPRCPRI subkeyword of the DATA-MOVER keyword of the COPY command.

DM-INCREMENTAL

Indicates whether a full volume Incremental FlashCopy relationship is to be established. This parameter applies only when DM-PGM is not specified or DM-PGM=ADRDSSU is specified. This parameter corresponds to the INCREMENTAL subkeyword of the DATA-MOVER keyword of the COPY command.

DM-MAXIMUM-SUBTASKS

Sets the absolute maximum number of subtasks that can be attached and used for volume snaps. This parameter applies only when DM-PGM=EMCSNAP is specified. This parameter corresponds to the MAXIMUM-SUBTASKS subkeyword of the DATA-MOVER keyword of the COPY command.

DM-NOCONCURRENT

Specifies whether the CONCURRENT option will be supplied to ADRDSSU. Valid values are YES | Y (the CONCURRENT option will not be supplied to ADRDSSU; the NOCONCURRENT keyword will be generated) or NO | N (the CONCURRENT option will be supplied to ADRDSSU; the NOCONCURRENT keyword will not be generated). This parameter applies only when DM-PGM is not specified or DM-PGM=ADRDSSU is specified. This parameter corresponds to the NOCONCURRENT subkeyword of the DATA-MOVER keyword of the COPY command.

CPARM RENAME parameters

These parameters are DB2 Cloning Tool Subsystem Cloning RENAME command parameters. More information about these parameters is available in topic Chapter 23, "DB2 Cloning Tool Subsystem Cloning commands," on page 337.

NOTRENAMED

Specifies the NOTRENAMED value to be used in the rename.

ISSUE-VCLOSE

Specifies the ISSUE-VCLOSE value to be used in the rename.

MAX-TASKS

Specifies the MAX-TASKS value to be used in the rename.

| **DATACLAS**

| Specifies the SMS DATACLAS to be used for all renamed data sets on SMS
| managed volumes.

| **DATACLAS-PAIRS**

| Specifies source/target pairs for DATACLAS.

| **DRIVEACS**

| Specifies that SMS class information for renamed data sets is to be derived by
| driving ACS routines. Possible values are YES | Y (generated the DRIVEACS
| keyword) or NO | N (do not generate the DRIVEACS keyword).

| **GDG-ALL-MIGRATED**

| Specifies the action to take if all the source GDG generations have been
| migrated.

| **GDG-EMPTY**

| Specifies the action to take if the source GDG base is empty.

| **GDG-MIGRATED**

| Specifies the action to take if some, but not all, of the source GDG generations
| have been migrated.

| **GDG-TAPE**

| Specifies the action to take if some, but not all, of the source GDG generations
| are on tape.

| **MISSINGUCAT**

| Specifies the disposition and return code to be generated for data sets found
| on a volume, where the data set name matches a rename mask, but the catalog
| back-pointer is not one of the source catalogs that are specified in the
| corresponding COPY command.

| **MGMTCLAS**

| Specifies the SMS MGMTCLAS to be used for all renamed data sets on SMS
| managed volumes.

| **MGMTCLAS-PAIRS**

| Specifies source/target pairs for MGMTCLAS.

| **ORPHANCATENTRY**

| Specifies the disposition and return code to be generated for data sets found in
| a catalog but one or more catalog volume cells are not in the list of volumes
| copied.

| **RECATALOG**

| Specifies that DB2 Cloning Tool can replace an existing catalog entry without
| considering it an error.

| **STORCLAS**

| Specifies the SMS STORCLAS to be used for all renamed data sets on SMS
| managed volumes.

| **STORCLAS-PAIRS**

| Specifies source/target pairs for STORCLAS.

| **RENAME-MASKS**

| (Required) Specifies the rename masks that will be used to rename the data
| sets. (required)

| **EXCLUDE-SRCNAME-MASKS**

| This parameter supplies a list of source data set names or masks that will not
| be renamed (excluded from renaming).

| **EXCLUDE-SRCNAME**

| This parameter specifies the return code that will be given for data sets that
| match entries in the EXCLUDE-SRCNAME-MASKS keyword.

| **RENAME-ERROR**

| This option specifies how processing proceeds when a RENAME error is
| encountered.

| **RENAME-AUDIT-LOG**

| Specifies whether an audit log of the data sets being renamed is to be created
| by RENAME volume processing.

| **RENAME-LIST**

| Specifies whether a list of the renamed data sets is to be produced by
| RENAME volume processing.

| **TEMPDSN**

| Specifies the disposition of temporary data sets and the return code to be
| generated if at least one occurrence is discovered.

| **UPDATE-IAM-ASSOCIATIONS**

| Specifies whether IAM data set associations are to be updated as part of
| RENAME processing.

| **VALIDATE-SMS-CLASSES**

| Specifies whether the SMS class names that are specified in the DATACLAS,
| DATACLAS-PAIRS, MGMTCLAS, MGMTCLAS-PAIRS, STORCLAS, and
| STORCLAS-PAIRS keywords will be validated as being defined to SMS.

| **CPARM parameters for pairs of subsystems or data sharing groups**

| The DB2-SUBSYSTEM-PAIR and DB2-GROUP-PAIR keywords denote the start of
| the cloning parameters for a pair of subsystems or data sharing groups.

| The DB2-SUBSYSTEM-PAIR keyword denotes a non-data sharing pair of a source
| DB2 SSID and a target DB2 SSID.

| The DB2-GROUP-PAIR keyword denotes a data sharing group pair of a source
| DB2 group and a target DB2 group. Use the SSID-PAIRS keyword to specify the
| source DB2 SSID and target DB2 SSID pairs of the members of the data sharing
| group. Also, specify the DATASHR-ATTR keyword to set the final data sharing
| attributes for the target group. In addition, you can specify the SSID-SURVIVING
| keyword to identify the target DB2 SSIDs that will survive for DATASHR-ATTR of
| FEWER or NONDS. For NONDS, there can be only one surviving SSID.

| **DB2-SUBSYSTEM-PAIR**

| Specifies the source DB2 SSID and its corresponding target DB2 SSID. This
| keyword marks the start of the keywords for a subsystem pair. This parameter
| is required for each non-data sharing DB2 being cloned.

| **DB2-GROUP-PAIR**

| Specifies the source DB2 group name and its corresponding target DB2 group
| name. This keyword marks the start of the keywords for a group pair. This
| parameter is required for each data sharing group that is being cloned.

| **SSID-PAIRS**

| Specifies the pairs of the source DB2 SSIDs and their corresponding target DB2
| SSIDs. This parameter is required for DB2-GROUP-PAIR.

|
| **DATASHR-ATTR**

| Specifies the data sharing attribute that the final target DB2 system will have.
| Valid values are SAME, FEWER, or NONDS. This parameter is required for
| DB2-GROUP-PAIR.

| **SSID-SURVIVING**

| Specifies the surviving target SSIDs. This parameter is required when
| DATASHR-ATTR=FEWER or NONDS is specified and applies only when
| DATASHR-ATTR is FEWER or NONDS.

| **DB2-HLQS**

| (Required) Specifies the source and target DB2 data set high level qualifiers.
| This parameter corresponds to the DB2-HLQS keyword of the DB2UPDATE
| command.

| **STOGROUPS**

| Specifies the source and target storage groups that are defined to the DB2
| subsystem. This parameter corresponds to the STOGROUPS keyword of the
| DB2UPDATE command.

| **ARCHIVE**

| If the DB2 archive logs are on the source DASD volumes, and are copied to the
| target volumes, this keyword specifies whether the data set names and
| VOLSERS of the DB2 archive logs in the BSDSs are to be changed to the target
| values. Valid values are YES | Y (generate the ARCHIVE keyword) or NO | N
| (do not generate the ARCHIVE keyword). This parameter corresponds to the
| ARCHIVE keyword of the DB2UPDATE command.

| **LISTSQL**

| Specifies whether the generated SQL should be included in the listing of the
| DB2SQL command. This parameter corresponds to the LISTSQL keyword of
| the DB2SQL command.

| **WLM-ENVIRONMENT-MASKS**

| Specifies the masks that will be used to rename the WLM environment names
| in SYSIBM.SYSROUTINES. This parameter corresponds to the
| WLM-ENVIRONMENT-MASKS keyword of the DB2SQL command.

| **DATACLAS-MASKS**

| Specifies the masks that will be used to rename the DATACLAS values in
| SYSIBM.SYSSTOGROUP. This parameter corresponds to the
| DATACLAS-MASKS keyword of the DB2SQL command.

| **MGMTCLAS-MASKS**

| Specifies the masks that will be used to rename the MGMTCLAS values in
| SYSIBM.SYSSTOGROUP. This parameter corresponds to the
| MGMTCLAS-MASKS keyword of the DB2SQL command.

| **STORCLAS-MASKS**

| Specifies the masks that will be used to rename the STORCLAS values in
| SYSIBM.SYSSTOGROUP. This parameter corresponds to the
| STORCLAS-MASKS keyword of the DB2SQL command.

| **CLEAN-SYSUTILX**

| Specifies whether SYSUTILX is to be cleaned out as part of the cloning. The
| cleaning out of SYSUTILX is done by using a job with the DB2UTILXCLEAN
| command. Valid values are YES | Y (generate the DB2UTILXCLEAN job) or
| NO | N (do not generate the DB2UTILXCLEAN job). If this parameter is not
| specified, the default is NO (to not clean out SYSUTILX).

CPARM BCSCLEAN command parameters

These parameters are DB2 Cloning Tool Subsystem Cloning BCSCLEAN command parameters. More information about these parameters is available in the topic Chapter 23, “DB2 Cloning Tool Subsystem Cloning commands,” on page 337.

CLEANUP-CATALOG-ORPHANS

Specifies whether BCSCLEAN should remove target catalog entries that belong to data sets that are on the target volumes but were not put there by the prior cloning. Valid values are YES | Y (generate the CLEANUP-CATALOG-ORPHANS keyword) or NO | N (do not generate the CLEANUP-CATALOG-ORPHANS keyword).

CLEANUP-CATALOG-DSNMASKS

This parameter supplies a list of data set names or masks of data sets that are not on any target volume, but are in a target user catalog and that should be deleted by BCSCLEAN.

CPARM RESTORE-FROM-DUMPTAPES command parameters

These parameters are DB2 Cloning Tool Subsystem Cloning RESTORE-FROM-DUMPTAPES command parameters. More information about these parameters is available in the topic Chapter 23, “DB2 Cloning Tool Subsystem Cloning commands,” on page 337.

RESTORE-DUMP-TAPE-UNIT

Specifies the tape unit to use for allocating the tape drives for the volume restores.

RESTORE-DUMP-MAX-TAPEDRIVES

Specifies the maximum number of tape drives to use for the volume restores. Valid values are 1-16, or specify an asterisk (*) to allow the RESTORE-FROM-DUMPTAPES command to set the maximum (set to 4 for this release).

RESTORE-DUMP-VARY-SCOPE

Specifies the scope of the vary commands against the target volumes. Valid values are GLOBAL or LOCAL.

Calling the stored procedure

This topic details the syntax for calling the DB2 Cloning Tool stored procedure.

Use the following SQL CALL statement to run the stored procedure:

```
EXEC SQL CALL CKZTOOLS.CLONE_SS( :TYPE,  
    :PPARMDSN, :PPARMMEM,  
    :SPARMDSN, :SPARMMEM,  
    :CPARMDSN, :CPARMMEM,  
    :SP_RC, :SP_MSGS)
```

The input parameters are:

TYPE

Refer to the following section for more information on this parameter; 10 characters are allowed.

PPARMDSN

The fully qualified data set name for the product parameter file; 44 characters are allowed.

| **PPARMMEM**

| If the product parameter file is a PDS, the member name that contains the
| product parameters; 8 characters are allowed.

| **SPARMDSN**

| The fully qualified data set name for the DB2 system parameter file; 44
| characters are allowed.

| **SPARMMEM**

| If the DB2 system parameter file is a PDS, the member name that contains the
| DB2 system parameters; 8 characters are allowed.

| **CPARMDSN**

| The fully qualified data set name for the cloning parameter file; 44 characters
| are allowed.

| **CPARMMEM**

| If the cloning parameter file is a PDS, the member name that contains the
| cloning parameters; 8 characters are allowed.

| The parameter files and their contents are described in detail in “Parameter files
| and parameter descriptions” on page 124.

| The output parameters are:

| **SP_RC**

| Returns an integer that represents the return code of the cloning job.

| **SP_MSGS**

| Contains one or more messages describing the outcome of the cloning job, in
| varchar format (length 1,331 bytes.)

| **TYPE parameters**

| The TYPE parameter identifies what function the stored procedure is to perform. It
| must be one of the following:

| **BUILD**

| Builds JCL and adds tasks to the DB2 administrative task scheduler. Sets up
| the environment for CLONE, RECLONE, and CLEAN.

| **BUILDJCL**

| Builds JCL only.

| **CLONE**

| Runs the initial cloning.

| **RECLONE**

| Stops the target DB2 systems, runs BCSCLEAN to clean up from the previous
| cloning and then runs the cloning.

| **REMOVE**

| Deletes all JCL and removes tasks from the DB2 administrative task scheduler.
| If CLONE or RECLONE have been done, a CLEAN should be done before
| REMOVE.

| **CLEAN**

| Stops the target DB2 systems and runs BCSCLEAN to clean up from the
| previous cloning. Can be used when the clone will no longer be used.

Verifying the cloning

Before performing the cloning, you should verify that the cloning will produce the desired results. Follow the steps in this topic to verify the cloning.

Procedure

1. Invoke the stored procedure with TYPE = BUILDJCL.
2. Check the stored procedure output. If SP_RC for BUILDJCL is not zero, use the messages in SP_MSGS to determine the changes that need to be made. Invoke the stored procedure with TYPE=REMOVE (depending on the BUILDJCL error REMOVE may give a non-zero return code), make changes to the parameter files, and then invoke the stored procedure with TYPE=BUILDJCL.
3. Once SP_RC for BUILDJCL equals zero, review the members in the product parameter data set to ensure that the generated jobs look as expected. If the generated jobs do not look as expected, invoke the stored procedure with TYPE=REMOVE , make changes to the parameter files, and then invoke the stored procedure with TYPE=BUILDJCL.
4. Once the generated JCL looks as expected, manually run the jobs in member name sequence.

Results

Once the jobs run successfully and produce the desired target DB2 system, the cloning has been verified.

What to do next

Before running the verified cloning, invoke the stored procedure with TYPE = REMOVE to remove the members in the product parameters data set. Then invoke the stored procedure with TYPE=BUILD. This will rebuild the JCL and add the cloning tasks to the DB2 administrative task scheduler.

JCL members generated by the stored procedure

This topic describes the JCL members generated by the DB2 Cloning Tool stored procedure.

The members put into the JCL library (JCL-DSN parameter) have a form of STnnn, where *nnn* starts at 001 and increments for each additional member.

- The first JCL library member(s) starting with ST001 are a DB2STOP of the target DB2 subsystem(s).
- Following the DB2STOP member(s) is a JCL library member for BCSCLEAN.
- Following the BCSCLEAN member are the cloning members, including DB2STOP/DB2SETLOG SUSPEND, COPY, DB2START/DB2SETLOG RESUME, and RENAME.

TYPE= CLEAN runs the members starting from ST001 (DB2STOP) through BCSCLEAN.

TYPE = CLONE runs the members starting after BCSCLEAN.

TYPE = RECLONE runs all the members.

For example, an online cloning of a typical DB2 subsystem generates the members in the following table:

Table 24. Stored procedure generated members for an online cloning

Member name	Description	Run for TYPE= CLEAN	Run for TYPE= CLONE	Run for TYPE= RECLONE
ST001	DB2STOP target DB2	Yes	No	Yes
ST002	BCSCLEAN	Yes	No	Yes
ST003	DB2SETLOG SUSPEND source DB2	No	Yes	Yes
ST004	COPY	No	Yes	Yes
ST005	DB2SETLOG RESUME source DB2	No	Yes	Yes
ST006	RENAME	No	Yes	Yes
ST007	DB2UPDATE target DB2	No	Yes	Yes
ST008	DB2START SPECIAL target DB2	No	Yes	Yes
ST009	DB2FIX DATABASES (DB2)	No	Yes	Yes
ST010	DB2SQL	No	Yes	Yes
ST011	DB2FIX DATABASES (APPLICATION)	No	Yes	Yes
ST012	DB2STOP target DB2	No	Yes	Yes
ST013	DB2START NORMAL target DB2	No	Yes	Yes

Running the verified cloning

Once the cloning has been verified, you can submit the jobs to clone the subsystem.

Invoke the stored procedure with TYPE = CLONE.

If SP_RC equals zero, the cloning has completed and the target DB2 system should be running.

If SP_RC is not zero, use the messages in SP_MSGS to determine the problem. It may also be necessary to look at the output of the cloning jobs that have run to determine the problem.

If you must change the generated JCL:

1. Invoke the stored procedure with TYPE=REMOVE.
2. Make changes to the parameter files.
3. Invoke the stored procedure with TYPE=BUILD.

Recloning a DB2 system

After a DB2 system has been cloned, you can reclone the system using these instructions.

Invoke the stored procedure with TYPE = RECLONE.

If SP_RC equals zero, the re-cloning has completed and the target DB2 system should be running.

If SP_RC is not zero, use the messages in SP_MSGS to determine the problem. It may also be necessary to look at the output of the cloning jobs that have run to determine the problem.

Stored procedure example: Cloning a non data sharing subsystem

This topic provides an example of cloning a non data sharing subsystem, including sample parameter files and stored procedure syntax.

The following is an example of cloning a non data sharing subsystem. The source DB2 SSID is DB2P and the target DB2 SSID is DB2T.

1. Set up the product parameter file as follows:

```
CKZINI = CKZ.V31.SCKZPARM(CKZINI)
SCKZLOAD = CKZ.V31.SCKZLOAD
SCKZPARM = CKZ.V31.SCKZPARM
```

2. Set up the DB2 system parameter file. A sample DB2 system parameter file might appear as follows:

```
* * * * *
* Source DB2 DB2P
* * * * *
SSID = DB2P
SDSNLOAD = DSN.Vxxx.SDSNLOAD
EXEC-SYSTEM = SYSA
*
* * * * *
* Target DB2 DB2T
* * * * *
SSID = DB2T
SDSNLOAD = DSN.Vxxx.SDSNLOAD
SDSNEXIT = DSN.DB2T.SDSNEXIT
SPECIAL-DSNZPARM = DB2TSPEC
EXEC-SYSTEM = SYST
BDS01 = DB2T.BSDS01
BDS02 = DB2T.BSDS02
SYSVCAT = DB2T
DDF-LOCATION = DB2TLOC
DDF-LUNAME = DB2TLU
DDF-PORT = 3001
DDF-REPORT = 3002
```

3. Set up the cloning parameter file. A sample cloning parameter file might appear as follows:

```
JCL-DSN = CKZ.CLONE1.JCL
STATUS-DSN = CKZ.CLONE1.STATUS
TASK-PREFIX = CLONE1
USERID = CKZUSER
PASSWORD = xxxxxxxx
JOB CARD1 = //CKZCLON1 JOB , 'CKZ CLONING1', CLASS=A, MSGCLASS=X
WORK-PREFIX = CKZ.CLONE1.WRK
CLONING-TYPE = ONLINE
*
* COPY parameters
FROM-STORAGEGROUP = DB2PSG
TO-STORAGEGROUP = DB2TSG
USERCATALOGS = ICF.DB2P.CATALOG ICF.DB2T.CATALOG
*
* RENAME parameters
MAX-TASKS = 10
STORCLAS-PAIRS = DB2PSC DB2TSC
```

```

RENAME-MASKS = DB2P.**    DB2T.**
*
* * * * *
* DB2 parameters for cloning DB2P to DB2T
* * * * *
DB2-SUBSYSTEM-PAIR = DB2P DB2T
DB2-HLQS = DB2P DB2T
WLM-ENVIRONMENT-MASKS = DB2P* DB2T*

```

4. Invoke the stored procedure.

```

EXEC SQL CALL CKZTOOLS.CLONE_SS( :TYPE,
      :PPARMDSN, :PPARMMEM,
      :SPARMDSN, :SPARMMEM,
      :CPARMDSN, :CPARMMEM,
      :SP_RC, :SP_MSGS)

```

Where:

```

PPARMDSN = CKZ.CLONE1.PARMS
PPARMMEM = PRODUCT
SPARMDSN = CKZ.CLONE1.PARMS
SPARMMEM = DB2SYS
CPARMDSN = CKZ.CLONE1.PARMS
CPARMMEM = CLONE1

```

5. Verify the cloning.
6. Run the verified cloning.

Stored procedure example: Cloning from a system-level backup

This topic provides an example of cloning a non data sharing subsystem from a DB2 system-level backup. The example includes sample parameter files and stored procedure syntax. The DB2 system-level backup is created by using the DB2 BACKUP SYSTEM utility.

The following is an example of cloning a non data sharing subsystem from a system-level backup. The source DB2 SSID is DB2P and the target DB2 SSID is DB2T.

1. Set up the product parameter file as follows:

```

CKZINI = CKZ.V31.SCKZPARAM(CKZINI)
SCKZLOAD = CKZ.V31.SCKZLOAD
SCKZPARAM = CKZ.V31.SCKZPARAM

```

2. Set up the DB2 system parameter file. A sample DB2 system parameter file might appear as follows:

```

* * * * *
* Source DB2 DB2P
* * * * *
SSID = DB2P
SDSNLOAD = DSN.Vxxx.SDSNLOAD
EXEC-SYSTEM = SYSA
*
* * * * *
* Target DB2 DB2T
* * * * *
SSID = DB2T
SDSNLOAD = DSN.Vxxx.SDSNLOAD
SDSNEXIT = DSN.DB2T.SDSNEXIT
SPECIAL-DSNZPARAM = DB2TSPEC
EXEC-SYSTEM = SYST
BDSO1 = DB2TLG.BDSO1
BDSO2 = DB2TLG.BDSO2
SYSVCAT = DB2T

```



```

DDF-LOCATION = DB2TLOC
DDF-LUNAME = DB2TLU
DDF-PORT = 3001
DDF-RESPORT = 3002

```

3. Set up the cloning parameter file. A sample cloning parameter file might appear as follows:

```

JCL-DSN = CKZ.CLONE1.JCL
STATUS-DSN = CKZ.CLONE1.STATUS
TASK-PREFIX = CLONE1
USERID = CKZUSER
PASSWORD = xxxxxxxx
JOB CARD1 = //CKZCLON1 JOB , 'CKZ CLONING1', CLASS=A, MSGCLASS=X
WORK-PREFIX = CKZ.CLONE1.WRK
CLONING-TYPE = ONLINE
*
*COPY parameters
SOURCE-VOLUMES = DB2SLB
SOURCE-TOKEN = LAST
SOURCE-LOCATION = DB2PLOC
SOURCE-STORAGEGROUP = DB2PSGLG DB2PSGDB
TO-STORAGEGROUP = DB2TSGLG DB2TSGDB
USERCATALOGS = ICF.DB2P.CATALOG.LG ICF.DB2T.CATALOG.LG
                ICF.DB2P.CATALOG.DB ICF.DB2T.CATALOG.DB
*
*RENAME parameters
MAX-TASKS = 10
STORCLAS-PAIRS = DB2PSC DB2TSC
RENAME-MASKS = DB2PLG.** DB2TLG.**
                DB2PDB.** DB2TDB.**
*
* * * * *
* DB2 parameters for cloning DB2P to DB2T
* * * * *
DB2-SUBSYSTEM-PAIR = DB2P DB2T
DB2-HLQS = DB2PDB DB2TDB
WLM-ENVIRONMENT-MASKS = DB2P* DB2T*

```

4. Invoke the stored procedure.

```

EXEC SQL CALL CKZTOOLS.CLONE_SS( :TYPE,
    :PPARMDSN, :PPARMMEM,
    :SPARMDSN, :SPARMMEM,
    :CPARMDSN, :CPARMMEM,
    :SP_RC, :SP_MSGS)

```

Where:

```

PPARMDSN = CKZ.CLONE1.PARMS
PPARMMEM = PRODUCT
SPARMDSN = CKZ.CLONE1.PARMS
SPARMMEM = DB2SYS
CPARMDSN = CKZ.CLONE1.PARMS
CPARMMEM = CLONE1

```

5. Verify the cloning.
6. Run the verified cloning.

Stored procedure example: Cloning from DB2 system-level backup dump tapes

This topic provides an example of cloning a non data sharing subsystem from DB2 system-level backup dump tapes. The example includes sample parameter files and stored procedure syntax. The DB2 system-level backup dump tapes are created by using the DB2 BACKUP SYSTEM utility with the DUMP keyword.

The following is an example of cloning a non data sharing subsystem from system-level backup dump tapes. The source DB2 SSID is DB2P and the target DB2 SSID is DB2T.

1. Set up the product parameter file as follows:

```
CKZINI = CKZ.V31.SCKZPARAM(CKZINI)
SCKZLOAD = CKZ.V31.SCKZLOAD
SCKZPARAM = CKZ.V31.SCKZPARAM
```

2. Set up the DB2 system parameter file. A sample DB2 system parameter file might appear as follows:

```
* * * * *
* Source DB2 DB2P
* * * * *
SSID = DB2P
SDSNLOAD = DSN.Vxxx.SDSNLOAD
EXEC-SYSTEM = SYSA
*
* * * * *
* Target DB2 DB2T
* * * * *
SSID = DB2T
SDSNLOAD = DSN.Vxxx.SDSNLOAD
SDSNEXIT = DSN.DB2T.SDSNEXIT
SPECIAL-DSNZPARAM = DB2TSPEC
EXEC-SYSTEM = SYST
BDSO1 = DB2TLG.BDSO1
BDSO2 = DB2TLG.BDSO2
SYSVCAT = DB2T
DDF-LOCATION = DB2TLOC
DDF-LUNAME = DB2TLU
DDF-PORT = 3001
DDF-REPORT = 3002
```

3. Set up the cloning parameter file. A sample cloning parameter file might appear as follows:

```
JCL-DSN = CKZ.CLONE1.JCL
STATUS-DSN = CKZ.CLONE1.STATUS
TASK-PREFIX = CLONE1
USERID = CKZUSER
PASSWORD = xxxxxxxx
JOB CARD1 = //CKZCLON1 JOB , 'CKZ CLONING1' , CLASS=A, MSGCLASS=X
WORK-PREFIX = CKZ.CLONE1.WRK
CLONING-TYPE = ONLINE
*
*COPY parameters
SOURCE-VOLUMES = DB2SLB
SOURCE-TOKEN = LAST
SOURCE-LOCATION = DB2PLOC
SOURCE-USE-DUMPTAPES = Y
RESTORE-DUMP-TAPE-UNIT = TAPE
RESTORE-DUMP-MAX-TAPEDRIVES = 2
RESTORE-DUMP-VARY-SCOPE = GLOBAL
SOURCE-STORAGEGROUP = DB2PSGLG DB2PSGDB
TO-STORAGEGROUP = DB2TSGLG DB2TSGDB
USERCATALOGS = ICF.DB2P.CATALOG.LG ICF.DB2T.CATALOG.LG
                ICF.DB2P.CATALOG.DB ICF.DB2T.CATALOG.DB
*
*RENAME parameters
MAX-TASKS = 10
STORCLAS-PAIRS = DB2PSC DB2TSC
RENAME-MASKS = DB2PLG.** DB2TLG.**
                DB2PDB.** DB2TDB.**
*
* * * * *
* DB2 parameters for cloning DB2P to DB2T
```

```

* * * * *
DB2-SUBSYSTEM-PAIR = DB2P DB2T
DB2-HLQS = DB2PDB DB2TDB
WLM-ENVIRONMENT-MASKS = DB2P* DB2T*

```

4. Invoke the stored procedure.

```

EXEC SQL CALL CKZTOOLS.CLONE_SS( :TYPE,
      :PPARMDSN, :PPARMMEM,
      :SPARMDSN, :SPARMMEM,
      :CPARMDSN, :CPARMMEM,
      :SP_RC, :SP_MSGS)

```

Where:

```

PPARMDSN = CKZ.CLONE1.PARMS
PPARMMEM = PRODUCT
SPARMDSN = CKZ.CLONE1.PARMS
SPARMMEM = DB2SYS
CPARMDSN = CKZ.CLONE1.PARMS
CPARMMEM = CLONE1

```

5. Verify the cloning.
6. Run the verified cloning.

Stored procedure example: Cloning from DB2 system-level backup dump tapes across sysplexes

This topic provides an example of cloning a non data sharing subsystem from DB2 system-level backup dump tapes across sysplexes. The example includes sample parameter files and stored procedure syntax. The DB2 system-level backup dump tapes are created by using the DB2 BACKUP SYSTEM utility with the DUMP keyword. When cloning across sysplexes, the DB2 SLB dump tapes that were created on the source system must be made available for use on the target system.

A DB2GETBACKINFO job must be run on the source system to extract the required information from HSM to identify the dump tapes to be used. The DB2GETBACKINFO job creates a backinfo data set that will be used as input to the cloning job on the target system. Before running the cloning on the target system, you must transfer the backinfo data set to the target system and specify the data set name of the transferred backinfo file in the SOURCE-BACKINFO-DSN parameter in the cloning parameter file.

The following is an example of cloning a non data sharing subsystem from system-level backup dump tapes across sysplexes. The source DB2 SSID is DB2P and the target DB2 SSID is DB2T.

1. Set up the product parameter file as follows:

```

CKZINI = CKZ.V31.SCKZPARAM(CKZINI)
SCKZLOAD = CKZ.V31.SCKZLOAD
SCKZPARAM = CKZ.V31.SCKZPARAM

```

2. Set up the DB2 system parameter file. A sample DB2 system parameter file might appear as follows:

```

* * * * *
* Source DB2 DB2P
* * * * *
SSID = DB2P
SDSNLOAD = DSN.Vxxx.SDSNLOAD
EXEC-SYSTEM = SYSA
*
* * * * *
* Target DB2 DB2T
* * * * *

```

```

SSID = DB2T
SDSNLOAD = DSN.Vxxx.SDSNLOAD
SDSNEXIT = DSN.DB2T.SDSNEXIT
SPECIAL-DSNZPARAM = DB2TSPEC
EXEC-SYSTEM = SYST
BSDS01 = DB2TLG.BSDS01
BSDS02 = DB2TLG.BSDS02
SYSVCAT = DB2T
DDF-LOCATION = DB2TLOC
DDF-LUNAME = DB2TLU
DDF-PORT = 3001
DDF-REPORT = 3002

```

3. Set up the cloning parameter file. A sample cloning parameter file might appear as follows:

```

JCL-DSN = CKZ.CLONE1.JCL
STATUS-DSN = CKZ.CLONE1.STATUS
TASK-PREFIX = CLONE1
USERID = CKZUSER
PASSWORD = xxxxxxxx
JOB CARD1 = //CKZCLON1 JOB , 'CKZ CLONING1', CLASS=A, MSGCLASS=X
WORK-PREFIX = CKZ.CLONE1.WRK
CLONING-TYPE = ONLINE
*
*COPY parameters
SOURCE-VOLUMES = DB2SLB
SOURCE-TOKEN = LAST
SOURCE-LOCATION = DB2PLOC
SOURCE-BACKINFO-DSN = CKZ.CLONE1.BACKINFO
SOURCE-USE-DUMPTAPES = Y
RESTORE-DUMP-TAPE-UNIT = TAPE
RESTORE-DUMP-MAX-TAPEDRIVES = 2
RESTORE-DUMP-VARY-SCOPE = GLOBAL
SOURCE-STORAGEGROUP = DB2PSGLG DB2PSGDB
TO-STORAGEGROUP = DB2TSGLG DB2TSGDB
USERCATALOGS = ICF.DB2P.CATALOG.LG ICF.DB2T.CATALOG.LG
                ICF.DB2P.CATALOG.DB ICF.DB2T.CATALOG.DB
*
*RENAME parameters
MAX-TASKS = 10
STORCLAS-PAIRS = DB2PSC DB2TSC
RENAME-MASKS = DB2PLG.** DB2TLG.**
                DB2PDB.** DB2TDB.**
*
* * * * *
* DB2 parameters for cloning DB2P to DB2T
* * * * *
DB2-SUBSYSTEM-PAIR = DB2P DB2T
DB2-HLQS = DB2PDB DB2TDB
WLM-ENVIRONMENT-MASKS = DB2P* DB2T*

```

4. Set up and run the DB2GETBACKINFO job. This job must be run manually on the source system to extract the required information from HSM to identify the dump tapes to be used. The following shows a sample DB2GETBACKINFO job:

```

//S1 EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DISP=SHR,DSN=CKZ.V31.SCKZLOAD
//CKZINI DD DISP=SHR,DSN=CKZ.V31.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//ABNLIGNR DD DUMMY
//BACKINFO DD DSN=CKZ.CLONE1.BACKINFO,
// DISP=(,CATLG),UNIT=SYSALLDA,
// SPACE=(CYL,(1,1))
//HSMLIST DD DSN=CKZ.CLONE1.WRK.HSMLIST,
// DISP=(,CATLG),UNIT=SYSALLDA,
// SPACE=(CYL,(1,1))
//CKZIN DD *

```

```

|         DB2GETBACKINFO           -
|         USE-DUMPTAPES           -
|         BACKINFO-DDN(BACKINFO)  -
|         WORK-DDN(HSMLIST)       -
|         LAST                     -
|         LOCATION(DB2PLOC)       -
|         USERCATALOGS(          -
|             ICF.DB2P.CATALOG.LG  -
|             ICF.DB2P.CATALOG.DB  -
|         )

```

/**

5. Transfer the backinfo data set created by DB2GETBACKINFO job to the target system (for example, using FTP or shared DASD).

6. Invoke the stored procedure.

```

EXEC SQL CALL CKZTOOLS.CLONE_SS( :TYPE,
    :PPARMDSN, :PPARMMEM,
    :SPARMDSN, :SPARMMEM,
    :CPARMDSN, :CPARMMEM,
    :SP_RC, :SP_MSGS)

```

Where:

```

PPARMDSN = CKZ.CLONE1.PARMS
PPARMMEM = PRODUCT
SPARMDSN = CKZ.CLONE1.PARMS
SPARMMEM = DB2SYS
CPARMDSN = CKZ.CLONE1.PARMS
CPARMMEM = CLONE1

```

7. Verify the cloning.

8. Run the verified cloning.

Chapter 11. DB2 Cloning Tool Table Space Cloning overview

DB2 Cloning Tool Table Space Cloning simplifies and automates the refresh of DB2 table spaces and index spaces. Paired with data set level fast data replication tools, data can easily be refreshed within minutes, instead of hours.

DB2 Cloning Tool Table Space Cloning's faster refresh rate alleviates troubleshooting time delays, and increases developer productivity. It is an offline utility that uses data set level fast replication tools, so it causes minimal disruption and fits into tight maintenance windows.

DB2 Cloning Tool Table Space Cloning makes it fast and easy for you to refresh DB2 test or quality assurance environments, troubleshoot production problems, and aid in development efforts because it:

- Uses high speed data set fast replication utilities, instead of traditional slow utilities
- Automates manual processes, such as object ID translation between source and target subsystems
- Uses TCP/IP to copy to subsystems not connected via CAF
- Uses an interface similar to the IBM LISTDEF facility to drastically reduce the learning curve
- Lets you select individual table spaces or an entire database
- Provides capability to exclude undesired table spaces and indexes
- Lets you select all table spaces in an RI relationship
- Easily select and migrate LOB tables and clone tables (for DB2 Version 9.1)
- Copies tables that have identity columns to another subsystem
- Beginning with DB2 Version 9.1, copies tables containing XML columns
- Includes the capability to mask column data. The changes are made based on masking rules that are enabled during the copy.
- Allows you to create the necessary jobs using ISPF interactive panels, if desired.

What does DB2 Cloning Tool Table Space Cloning do?

Cloning is the act of replicating data, making it accessible, and then using the replica in lieu of the original data for other purposes. Data set fast replication tools clone DB2 table spaces and index spaces quickly, and DB2 Cloning Tool Table Space Cloning makes the DB2 table spaces and index spaces accessible.

Fast-replicate and copy products used by DB2 Cloning Tool Table Space Cloning

DB2 Cloning Tool Table Space Cloning will make a usable clone from DB2 table spaces and index spaces created with any type of data set level fast replication utility or slow copy mechanism. DB2 Cloning Tool Table Space Cloning uses high-speed data set level fast-replicate utilities to do the copy, instead of using traditional utilities such as IBM DSN1COPY and UNLOAD/LOAD; but also works with slow copies such as DFSMSdss copy, FDR, etc.

- IBM FlashCopy
- STK SnapShot

- EMC TimeFinder/Clone Mainframe Snap Facility's data set level support

For IBM FlashCopy and STK SnapShot, DB2 Cloning Tool Table Space Cloning initiates the data set copies by executing DFSMSdss program, ADRDSSU, "under the covers." For EMC TimeFinder/Clone data set level support, DB2 Cloning Tool Table Space Cloning uses an EMC API.

IBM FlashCopy, STK SnapShot, and EMC TimeFinder/Clone allow the creation of what appears to be a copy of a data set almost instantaneously. After a copy is completed, DB2 Cloning Tool Table Space Cloning turns the resulting replicas into usable clones.

For any other cloning mechanisms, you must create the replica, using a list compiled by DB2 Cloning Tool Table Space Cloning, of data sets to copy. DB2 Cloning Tool Table Space Cloning then turns the copied data sets into usable clones.

Why you should use DB2 Cloning Tool Table Space Cloning

DB2 Cloning Tool Table Space Cloning saves time, money, and productivity in the following ways:

- Automating labor-intensive tasks increases your efficiency and reduces the potential for error
- Quicker throughput and turnaround time keeps your data available for use
- Reduces refresh costs because it is faster than traditional utilities, and eliminates time intensive manual research
- Less personnel time required to refresh DB2 table spaces and index spaces
- Allows you to manage growing environments using existing staff and within shorter time windows
- Simplifies data refresh by automating manual steps required when using traditional utilities

Can I refresh DB2 table spaces and index spaces without DB2 Cloning Tool Table Space Cloning?

DB2 table spaces and index spaces can be refreshed without using DB2 Cloning Tool Table Space Cloning by using DSN1COPY or UNLOAD/LOAD utilities, however:

- DSN1COPY requires static JCL and control parameters and does not allow for adding new DB2 extents, adding new table spaces or index spaces, or dropping existing ones. Object ID translation parameters require you to do painstaking manual research and maintenance.
- DB2 UNLOAD/LOAD can require a significant amount of time before large cloned data sets are available for use. The VSAM objects on the target side can require more space than on the source side due to FREESPACE and FREEPAGE assignments; therefore, the LOAD utility could abend and require a manual increase of space for a target table space or index space. In addition, extra time is needed to rerun the LOAD process.

Using DB2 Cloning Tool Table Space Cloning, DB2 table spaces and index spaces can be refreshed very quickly if using a fast replication utility. DB2 Cloning Tool Table Space Cloning dynamically adjusts to new or dropped table spaces and index spaces, performs automatic object ID translation, and there are no unanticipated size changes between the source and target table spaces and index spaces.

DB2 Cloning Tool Table Space Cloning features and benefits

DB2 Cloning Tool Table Space Cloning offers the following features and benefits.

- Clones DB2 table spaces and index spaces
- Is extremely fast, providing quick refresh
- Automatically invokes FlashCopy, SnapShot, or TimeFinder/Clone data set level support
- Works with both data set level fast replication utilities, and slow copy utilities
- Automates the object ID translation between the source and target DB2 table spaces and index spaces by removing the labor intensive manual research required with traditional utilities
- Includes extents automatically
- Ensures there will not be an incompatibility between the source and target table spaces and index spaces - if detected, DB2 Cloning Tool Table Space Cloning will not copy it, and provides a warning
- Uses a simple LISTDEF-like facility for ease-of-use that DBAs are already familiar with
- Supports RI relationships, LOBS, and identity columns
- Does not require a backup step
- Is not prone to copy failures due to table expansion - with DB2 Cloning Tool Table Space Cloning the target takes up the same amount of space as the source
- Copies the table spaces and index spaces outside of DB2 for faster refresh and less resource usage
- Automatically stops and starts table spaces and index spaces
- Provides integrity between the tables and indexes by stopping table spaces before index spaces
- Removes the requirement to rebuild indexes
- Optionally resets the table space time stamp
- Easier to use from start to finish than traditional utilities
- Replaces labor intensive tasks with automation to reduce errors and increase efficiency
- Reduces the need to update utility JCL in production libraries
- Simulate mode for the COPY command
- Can copy DB2 tables containing XML column data
- Optionally, when using DSS or the EMC API, DB2 Cloning Tool Table Space Cloning can tolerate enqueue failures and not stop the source spaces
- Can copy to new creator ID and/or object name

DB2 Cloning Tool Table Space Cloning terminology - DB2 objects vs. VSAM objects

DB2 Cloning Tool Table Space Cloning uses the term "objects" in two ways.

DB2 entities, such as tables, table spaces, indexes, index spaces, and applications, may be referred to as "objects", or by the specific object type, such as "table spaces" or "index spaces".

External VSAM data sets that contain table spaces and index spaces are referred to as VSAM objects.

Use of the U.S.A. EBCDIC code set

DB2 Cloning Tool uses the U.S.A. EBCDIC code set for specification and display of EBCDIC characters and for the extended ACS masking characters used for filtering.

If the code tables used by your installation are different, then you need to enter the EBCDIC character peculiar to your code tables that results in the binary value for the EBCDIC character specified in the product manuals.

- Data supplied as input to batch programs or input to ISPF panels:

For product code shipped in binary, when specifying input where the product takes special action based on specific characters, you must enter the EBCDIC character peculiar to your code tables that results in the binary value for the EBCDIC character specified in the product manuals, according to the U.S.A. EBCDIC code set.

For example, if an exclamation mark (!) is called for, and your code tables do not translate the ! character to a hexadecimal 5A, you must enter the character that your code table will translate to a 5A.

- Distributed ISPF panels:

Do not change distributed ISPF panels. Program code may reference ISPF panel attribute bytes. A panel change that affects an attribute byte may cause a program error.

- Product output:

Depictions of product output shown in the product manuals are based on the U.S.A. EBCDIC code set. Actual output may vary if your EBCDIC code tables are different.

- Extended ACS masking characters:

Your installation may need to specify different masking characters to achieve the desired result if your code tables are different from the U.S.A. EBCDIC code set. For more information about extended ACS masking characters, see "Filtering pattern masks" on page 23.

Chapter 12. Planning for copying DB2 table spaces by data set

Before attempting to actually use DB2 Cloning Tool Table Space Cloning, some planning and decision-making should take place. This topic discusses those things that need to be considered.

Considerations for in-progress read/write activity or DB2 utilities

DB2 utility activity or other read/write activity can affect DB2 Cloning Tool Table Space Cloning processes.

DB2 table spaces and index spaces being copied should not have a DB2 utility in progress. In addition, there should not be any read/write activity against source or target table spaces or index spaces to be copied (unless the FUZZY copy option is chosen).

DB2 Cloning Tool Table Space Cloning issues a DB2 STOP command to the specific source and target DB2 table spaces and index spaces being copied. The DB2 STOP command waits with a pending status (STOPP) until the table spaces and index spaces are no longer in use. There is no way to force activity to stop. Only a successful STOP command will deallocate the associated VSAM data set from DBM1. If the table spaces or index spaces are in use, DB2 Cloning Tool Table Space Cloning waits a user-specified number of seconds for the STOP command to complete successfully.

Considerations for the DB2 Cloning Tool Table Space Cloning cloning process

The following rules must be taken into consideration when using DB2 Cloning Tool Table Space Cloning.

Subsystem copy rules

Table spaces and indexes spaces can be copied to the same DB2 subsystem or a different DB2 subsystem.

Attention: Copying of table spaces and index spaces between subsystems at different DB2 version levels is not recommended. Data may be missing from the target spaces or other unpredictable results can occur.

The DB2 Cloning Tool Table Space Cloning OBJECT-TRANSLATE keyword allows the copying of table spaces and index spaces to the same subsystem. Once the source table spaces and index spaces have been determined, the same database and space name are used for the target data sets unless object translation is specified. If object translation is specified, DB2 Cloning Tool Table Space Cloning attempts to use the supplied OBJXLAT commands to map the existing database and space names into new target data sets.

When processing object translation, DB2 Cloning Tool Table Space Cloning does not verify that the resulting target index will correctly point to the resulting target table. However, DB2 Cloning Tool Table Space Cloning does verify that the table

spaces and index spaces are compatible from the source to the target. Object translation should be carefully specified to ensure proper results.

Renaming table spaces and index spaces

Table spaces and index spaces cannot be renamed because DB2 doesn't support it.

A rename of a DB2 table space or index space can only be accomplished by using the DB2 DROP and CREATE commands. A DB2 DROP deletes both the VSAM data set and the DB2 catalog information concerning the DB2 table space or index space. However, a source table space or index space can be copied to an existing target table space or index space with a different name.

Identity columns

DB2 Cloning Tool Table Space Cloning can update the sequence numbers for identity columns in the DB2 catalog.

DB2 Cloning Tool Table Space Cloning issues ALTER TABLE SQL to correct identity column values.

LOBs, BLOBs, CLOBs, and DBCLOBs

When copying a large object (LOB) table, both the table space containing the LOB column and the table space containing the auxiliary table with the LOB data must be copied at the same time using the LISTDEF ALL command.

The target object or objects may not be accessible if BASE or LOB LISTDEF keywords are specified.

XML considerations

This topic describes the considerations for cloning table spaces containing XML column data.

Cloning tables containing XML column data

DB2 Version 9.1 provides fully integrated storage of XML data in the DB2 database system. The XML column data type is provided for storing XML data in DB2 tables.

The following steps are required when the source and target subsystems are different.

1. Add the XMLSTRING-DDN subcommand to the COPY command in the source job. This subcommand is used to pass the source subsystem XMLSTRINGS catalog table contents from the source job to the target job.
2. Add an XMLSTR DD to the source job for the DDN pointed to in step 1.
3. Add the identical XMLSTR DD in the source job to the target job.
4. Add the following DD to the target job. This DD contains the DDL to create an XML table for adding strings to SYSIBM.SYSXMLSTRINGS using an SQL INSERT. The data set pointed to by the CKZCRXML DD must have an LRECL of 80 and RECFM of FB.

```
CKZCRXML DD DISP=SHR,DSN=h1q.CKZMDDL(member_name)
```

DB2 Cloning Tool Table Space Cloning will use the DDL to create objects for the duration of the target job and parse the member to get the database, table creator, table name, bigint column name and XML column name. The database

is dropped when the target job completes XML processing. See SCKZJCL (CKZXMDDL) for a sample of the object create DDL.

XML string processing

DB2 Cloning Tool Table Space Cloning must add string IDs to the target SYSIBM.SYSXMLSTRINGS catalog table when the source and target subsystems are different and the XML column(s) contain string IDs in the source table space that are not in the target table space or that are mapped to different string IDs. The XML table object created by CKZXMDDL is used to INSERT XML data. DB2 then adds strings into the target catalog table SYSIBM.SYSXMLSTRINGS. The string IDs are translated from the source to the target value within the data pages of XML columns.

Other XML considerations

- When copying on the same subsystem and using object translation to specify an index space for DB2-built XML indexes, an object translate command must be specified even if the indexspace names are the same. For example, add these object translate commands:

```
INDEXSPACE,IRDOCIDZ,IRDOCIDZ, -  
INDEXSPACE,IRNODEID,IRNODEID, -
```

If you are certain that all indexspace names are the same from source to target, you can use the following command:

```
INDEXSPACE,%,%, -
```

- If the object translate commands are not added, CKZ54701W appears twice and indicates that no target object was found for the database.indexspace.
- When copying a table with an XML column, both the table space containing the XML column and the table space containing the XML data should be copied at the same time using the LISTDEF ALL command. If you use only BASE or XML LISTDEF keywords, the target object or objects may not be accessible.

DDF and TCP/IP considerations

If the target subsystem is on a different LPAR than the source subsystem, DB2 Cloning Tool Table Space Cloning requires that a TCP/IP connection be available from DB2 Cloning Tool Table Space Cloning to a DB2 Cloning Tool Table Space Cloning TCP/IP server on the remote z/OS system.

If a DDF connection is available, it is used to allow DB2 Cloning Tool Table Space Cloning to gather information from the target DB2 catalog. If no DDF connection is available, TCP/IP is used to access the target catalog. TCP/IP is used to issue DB2 commands. Communication with the target is how DB2 Cloning Tool Table Space Cloning can verify compatibility between source and target DB2 table spaces and index spaces. This communication also allows it to acquire and translate the internal object identifiers between the source and the target DB2. For TCP/IP communication, DB2 Cloning Tool Table Space Cloning provides server job CKZTCPS. This job is available in the product JCL library.

Note: Use SET REMOTE-CONNECT-TYPE to specify CAF, DDF or TCP/IP. Furthermore, if using DDF, a TCP/IP connection is required.

Object attributes

The attributes and contents of the source DB2 table space and index space data sets must not conflict with the attributes of the target DB2 table space and index space data sets.

For example, they must have the same object types, buffer pool sizes, DSSIZE, and use the same code page (CCSID).

The table columns and column attributes of the source and target tables must be identical. Other object attributes that must be identical are listed in the following table:

Table 25. Object attributes that must be identical between source and target objects

Object type	Attributes that must be identical for source and target object
Database	None
Table space	DSSIZE, PARTITIONS, PGSIZE, NTABLES, SEGSIZE, TYPE, CCSID, ORGANIZATIONTYPE
Partitioned table space	COMPRESS, FORMAT
Table	COLCOUNT, EDPROC, HASHKEYCOLUMNS
Column	COLNO, COLTYPE, LENGTH, SCALE, DEFAULT, FLDPROC, SOURCETYPEID, SEQTYPE, CYCLE, CACHE, START, MINVALUE, MAXVALUE, INCREMENT, HASHKEY_COLSEQ
Index	UNIQUERULE, COLCOUNT, CLUSTER, PGSIZE, INDEXTYPE, PIECESIZE, HASH, SPARSE, UNIQUE_COUNT, COLNO, COLSEQ, ORDERING; also the number of partitions in the index must be identical.
Partitioning index	LIMITKEY

In general, differences that affect the layout of the data or the data set prevent successful synchronization between source and target table spaces and index spaces. Attributes that affect only how DB2 processes the data, such as LOCKRULE, can be different.

If DB2 Cloning Tool Table Space Cloning detects a mismatch, a warning message with a return code of 4 is generated in the output. You can override the return codes that are assigned to specific mismatch types by using the OBJECT-MISMATCH-RETURN-CODE keyword in the COPY command. The return code and the warning message that is issued can be 0 (informational message), 4 (warning message), or 8 (error message).

APF authorization

The DB2 Cloning Tool Table Space Cloning LOAD library must be APF authorized.

This is included as a configuration step in “APF authorizing load libraries” on page 27.

Ensure RACF and ACF authorities

Before cloning, you should ensure that all proper RACF and ACF permissions and privileges are in place.

Refer to “Verify that your environment meets security requirements” on page 19 for information about required permissions and privileges for DB2 Cloning Tool Table Space Cloning.

Missing target table spaces and index spaces

DB2 Cloning Tool Table Space Cloning uses a single target high level qualifier for all VSAM objects associated with DB2 table spaces and index spaces missing from the target DB2 catalog. It will not be able to acquire the VCAT (high level qualifier) for the missing table spaces and index spaces from the DB2 catalog. DB2 Cloning Tool Table Space Cloning will use the DEFVCAT (default VCAT) parameter supplied in the COPY command for all such VSAM data sets.

Note: If a DB2 table space or index space is created on the target, after the VSAM object is copied from the source, there can be a problem: When DB2 is instructed to create the target table space or index space, it will place the target internal object identifiers inside the existing VSAM object. That immediately makes any existing data inside the VSAM object permanently inaccessible. The solution is to make the name of target VSAM object unrecognizable by DB2. Allow DB2 to create a new VSAM object, then delete that object and rename the target VSAM object to make it recognizable to DB2.

Considerations for target objects created using DEFINE NO

If a DB2 table space or index space is created on the target with DEFINE NO, the data set for the object will not exist until a row is inserted into the object. The target DB2 subsystem must be forced to create target data sets for the “DEFINE NO” objects.

If the target DB2 subsystem does not create the target data sets, DB2 will never access the data sets cloned from the source DB2. When the target data set does not exist, DB2 Cloning Tool Table Space Cloning will copy the object as if the target object does not exist. In this case, you may force the target DB2 subsystem to create the target data sets either before the source and target jobs execute or after they execute.

If the target data set for the “DEFINE NO” object is created before the source and target jobs execute:

1. Execute the source job.
2. After the source job completes successfully, execute the target job.

If the target data set for the “DEFINE NO” object will be created after the source and target jobs execute:

1. Submit the source job. The source job will target output data sets with the fifth node qualifier of *. F0001.* (i.e. HLQ.DSNDB*.DBNAME.TSNAME.F0001.Annn). This is done so that when the target DB2 attempts to create an *.I0001.* data set, it will not find the one copied from the source already in existence. If DB2 found an *.I0001.* data set, it would initialize the existing data set and all data would become inaccessible.
2. Submit the target job. The target job will update object IDs in the *.F0001.* data set(s) to match those in the target catalog. It will also automatically start the target object(s). To assist you in renaming the data sets, IDCAMS parameters are written to an optional IDCAMS-DDN file. These parameters may be used to rename the *.F0001.* data set(s) to *.I0001.* data set(s).
3. You must then complete the following steps:
 - a. Insert a row into the target table. If the table is partitioned then a row must be inserted into the desired partition(s).
 - b. Stop the target object(s). This prevents enqueue conflicts with the IDCAMS job over the *.I0001.* data sets.

- c. Submit a job using the IDCAMS parameters to rename the *.F0001.* data set(s) to *.I0001.* data set(s). This job will delete the *.I0001.* data sets created by DB2 and then rename the *.F0001.* data sets to *.I0001.* data sets.
- d. Start the target object(s).

Note: If the source job is submitted with the DATA-MOVER(PGM(NONE)) keyword, the -STOP parameters for all target objects will be written to the optional STOP-TARGET-DDN file. These parameters may then be copied and edited to create -START parameters for the target objects.

The easiest way to copy using DB2 Cloning Tool Table Space Cloning is to define the target object with a minimum size. Then when DB2 Cloning Tool Table Space Cloning copies the source data, the old data set is deleted and a new larger data set is allocated. No additional processing would be required.

If it is not known for certain whether any target objects were created with DEFINE NO and do not have a data set created by DB2, submit a DB2 Cloning Tool Table Space Cloning source job with COPY-IF-NO-DB2-TARGET-OBJECTS(N) and PGM(NONE). All target table space and index spaces without a data set will be listed in the output with a warning message.

Non-SMS managed volumes

If DB2 Cloning Tool Table Space Cloning is controlling the copy process and the target volumes are non-SMS managed, the volsers needed for original data set(s) and the target data set(s) allocation may be passed to ADRDSSU via the HLQDDDF command as defined in the DB2 Cloning Tool Table Space Cloning source job, as shown in this topic.

The HLQDDDF DIRECTION(IN) parameter passes the volsers for input volumes using the DD names from the source job JCL:

```
MYINDD DD DISP=SHR,UNIT=3390,VOL=SER=(JM505F,JM515F)
MYINDD2 DD DISP=SHR,UNIT=3390,VOL=SER=(JM525F,JM535F)
```

Similarly, DIRECTION(OUT) passes volsers for output volumes using the DD names from the source job JCL:

```
MYOUTDD DD DISP=OLD,UNIT=SYSDA,VOL=SER=(JM505A,JM515A)
MYOUTDD2 DD DISP=OLD,UNIT=3390,VOL=SER=(JM525A,JM535A)
```

The command HLQDDDF may occur multiple times in the CKZIN DD, and keyword DDNAME can pass multiple DDnames:

```
CKZIN DD *
HLQDDDF HLQNAME(FDRABR1) DIRECTION(IN) DDNAME(MYINDD,MYINDD2)
HLQDDDF HLQNAME(FDRABR2) DIRECTION(OUT) DDNAME(MYOUTDD,MYOUTDD2)
```

Space processing order

DB2 Cloning Tool Table Space Cloning always processes table spaces first, followed by index spaces.

How DB2 Cloning Tool Table Space Cloning starts and stops table spaces and index spaces

DB2 Cloning Tool Table Space Cloning starts and stops source and target table spaces and index spaces depending on certain parameter settings.

Source table spaces and index spaces

For source table spaces and index spaces, the DB2 Cloning Tool Table Space Cloning COPY command keywords FUZZY-COPY and AUTO-START-SOURCE-SPACE control whether source table spaces and index spaces are stopped and started before and after the copy.

Note: DB2 stops and starts are done only if the COPY command keyword DATA-MOVER is set to PGM(ADRDSSU) or PGM(EMCAPI). DATA-MOVER PGM(NONE) does not start or stop any DB2 objects.

- If COPY command keyword FUZZY-COPY(Y) is in effect, no stops or starts will be applied to the source table spaces and index spaces. Table spaces and index spaces are copied in their current state.
- If COPY command keyword FUZZY-COPY(N) has been specified, before any copies begin, all source table spaces and index spaces are stopped. After all copies are completed, all source spaces are started according to the AUTO-START-SOURCE-SPACE keyword. COPY command keyword SIMULATE(A) or SIMULATE(N) will both result in source table spaces and index spaces being stopped and started when FUZZY-COPY(N) is in effect.

If an error occurs when attempting to stop a source space, DB2 Cloning Tool Table Space Cloning tries to restart all the source spaces that were stopped to their initial status.

Target table spaces and index spaces

For target table spaces and index spaces, the DB2 Cloning Tool Table Space Cloning COPY command keywords or INI tokens AUTO-STOP-TARGET-SPACE and AUTO-START-TARGET-SPACE, and the COPY command keyword SIMULATE (or SIM), control stopping and starting of target table spaces and index spaces.

- COPY command keyword or INI token AUTO-STOP-TARGET-SPACE: If set to Y, the source job stops the target table spaces and index spaces.
 - If SIM(A) is in effect, the source job restarts the target table spaces and index spaces, as the target job will not run until the table spaces and index spaces are copied with SIM(N).
 - If SIM(N), the table spaces and index spaces remain stopped.

Note: If AUTO-STOP-TARGET-SPACE(Y) is specified, then the target table spaces and index spaces will be stopped even if FUZZY-COPY(Y) is in effect.

- COPY command keyword or INI token AUTO-START-TARGET-SPACE: This keyword or INI token determines whether the DB2 Cloning Tool Table Space Cloning target job starts the spaces after SYNCDB2 processing is completed on the target DB2 subsystem. If AUTO-START-TARGET-SPACE(Y), each space is started as soon as its SYNCDB2 processing is completed.

If an error occurs while attempting to stop a target space:

- If SIM(A) is in effect, DB2 Cloning Tool Table Space Cloning tries to restart all the target spaces that were started when DB2 Cloning Tool Table Space Cloning was invoked.
- If SIM(N) is in effect, the target spaces will not be restored to their pre-DB2 Cloning Tool Table Space Cloning status. DB2 Cloning Tool Table Space Cloning assumes the user will stop the target table spaces and index spaces in STOPP (stop pending) outside of DB2 Cloning Tool Table Space Cloning and then rerun the source job.

When target spaces are restarted due to a STOPP error, DB2 Cloning Tool Table Space Cloning remembers the status of each table space and index space [RO (read only) or RW (read-write)] and issues the correct START command. Note that in the target job all target spaces are started RW.

If table spaces and index spaces do not exist on the target, no commands are issued.

Commands needed to stop and start table spaces and index spaces can be obtained by running the DB2 Cloning Tool Table Space Cloning source job with PGM(NONE). This can be useful if manual commands must be entered. Specify STOP-TARGET-DDN, STOP-SOURCE-DDN and START-SOURCE-DDN in the COPY command.

Message output

If warning or error messages are being output and are difficult to find in the CKZPRINT data set, you can route the warning and error messages to a separate DD.

Add the CKZERROR DD to the source, target and/or TCP/IP server jobs. When this DD is present, all warning and error messages will be output to this DD, as well as to CKZPRINT. For example:

```
//CKZERROR DD SYSOUT=*
```

Considerations for DB2 Version 9.1 and later

The following considerations apply when using DB2 Cloning Tool Table Space Cloning to clone table spaces and index spaces on a DB2 Version 9.1 or later subsystem.

Support for clone tables

DB2 Version 9.1 introduced the clone table, which is a copy of a table that resides in the same table space and that has the same attributes, structure, and data as the original base table. After you create a clone table, you can insert or load data into the clone table and exchange the clone table name with the base table name.

If a table is cloned, DB2 Cloning Tool Table Space Cloning always copies the clone table with its base table. DB2 Cloning Tool Table Space Cloning does not allow a base table or its clone to be copied to a table space that is not cloned, nor does it allow you to copy a non-cloned table to a base or cloned table space.

If the table space or index space does not exist on the target subsystem, DB2 Cloning Tool Table Space Cloning can copy both the base table and its clone table. DB2 Cloning Tool Table Space Cloning assigns the instance numbers in the underlying target VSAM data sets as F0001 for base tables, and F0002 for clone tables.

If desired, you can use the DB2 Cloning Tool Table Space Cloning LISTDEF command to selectively copy either only tables that have been cloned, or only those tables that have not been cloned.

Support for expanded index page sizes

DB2 Version 9.1 offers expanded index page sizes of 8 KB, 16 KB, and 32 KB. DB2 Cloning Tool Table Space Cloning can copy these expanded index page sizes.

Considerations for partition-by-growth table spaces

DB2 Version 9.1 introduced partitioning according to data growth. This enables segmented tables to be partitioned as they grow. DB2 Cloning Tool Table Space Cloning can copy partition-by-growth table spaces if the number of source partitions is equal to or less than the number of target partitions. However, if the number of source partitions is greater than the number of target partitions, DB2 Cloning Tool Table Space Cloning will create a DSN on the target for each partition that does not exist, using .F0001 as a data set name qualifier. This data set, and therefore the extra partitions, will not be usable on the target. If the number of partitions is greater on the source, you must either create the extra partitions(s) on the target using the DB2 UNLOAD and LOAD utilities before initiating the cloning process, or use the EXTEND-TARGET-PBG-TABLESPACE keyword of the COPY command to add the partitions during the cloning process.

Considerations for reordered row format in DB2 Version 9.1 NFM and later

Reordered row format (RRF) is new with DB2 Version 9.1 NFM. It results in variable-length columns being placed at the end of a row. RRF is not compatible with objects that have been migrated from DB2 V8. Those objects will be in basic row format (BRF) until a REORG or LOAD REPLACE is run on the object under DB2 Version 9.1 NFM.

Newly created table spaces under DB2 Version 9.1 NFM are put in RRF. If a source BRF table space is copied to a target RRF table space (or vice versa), in most cases the target table(s) will not be accessible. DB2 Cloning Tool Table Space Cloning will issue a warning if the row formats do not match. Thus, if running DB2 Cloning Tool Table Space Cloning on DB2 Version 9.1 NFM or later subsystem, and new objects are added to LISTDEF, run the source job using DATA-MOVER(NONE) and SIM(N). All object incompatibilities will result in warning and/or error messages. REORG the table spaces in BRF and rerun the DB2 Cloning Tool Table Space Cloning source job to ensure there are no more object incompatibilities.

Considerations for data set renaming

This topic discusses considerations for renaming data sets.

Qualifiers cannot be added or removed

When cloning DB2 table spaces and index spaces, qualifiers cannot be added or removed because of DB2 naming standard requirements. If a new database exists on the target, the new database name will be used in the VSAM data set.

Target data set ICF catalog aliases

Users are responsible for creating ICF catalog aliases, if needed, for new target data set names. DB2 Cloning Tool Table Space Cloning gets the target high level qualifier from the VCAT in the DB2 catalog. There must be an ALIAS in the ICF catalog to match each VCAT in the DB2 catalog to enable the rename and catalog.

If the target table spaces and index spaces do not exist, then there must also be an ALIAS to match the DEFVCAT specified in the COPY command.

Considerations for using FUZZY-COPY

This topic discusses considerations for cloning data sets using FUZZY-COPY.

If COPY command keyword FUZZY-COPY(Y) is specified, then DB2 Cloning Tool Table Space Cloning will not stop the source table spaces and index spaces. However, **we do not recommend this**, because if the table spaces and index spaces are in RW status, there are data integrity issues.

To safely use FUZZY-COPY (Y), do the following:

1. Start all table spaces to be copied in read only (RO) status.
2. Run a QUIESCE with WRITE(YES) on all table spaces to be copied.
3. Run the source job.
4. Start all table spaces in read-write (RW) status.

Considerations for generating target object DDL using PROCESS-DDL

DB2 Cloning Tool Table Space Cloning can optionally generate and execute DDL to be used for creating non-existent target objects. The PROCESS-DDL parameter of the COPY command generates DDL for databases, table spaces, tables, and indexes. LOB and XML spaces are supported. DB2 Cloning Tool Table Space Cloning can save the generated source object DDL to a data set, or can execute existing DDL from an input data set.

How LISTDEFs control DDL generation

As LISTDEF statements select objects to be processed, they also select the DDL to generate.

When missing target objects require DDL to be executed on the target, the DDL is built only for missing objects that must have a target data set for the copy from the source and objects they are dependent on.

When DDL for all source objects is to be created but not run, missing target objects are not relevant. In this case, LISTDEF is used differently to select objects that require DDL. DDL generation starts with all the table spaces that are directly or indirectly referenced in LISTDEF statements. This list of table spaces is referred to as the *object set*. All databases that contain table spaces in the object set have DDL generated. However, not all table spaces in each database have DDL generated. The table space must be in the object set selected by LISTDEF.

There must be at least one table space specified in the LISTDEF to generate DDL. A LISTDEF command with only indexes results in an error. All indexes must have a corresponding table space in the LISTDEF or be RI related (if specified). Indexes must also be specified in LISTDEF, either explicitly or implicitly (using ALWAYS-COPY-INDEXSPACES(Y)).

DDL generation and object translation

Use object translation to change object names from the source to the target. Use PROCESS-DDL to generate and execute DDL to create missing target objects.

Whenever object translation is performed, DB2 Cloning Tool Table Space Cloning automatically generates PROCESS-DDL attribute change parameters. The generated syntax is included in the job log data set. For example: Assume the object translate card DATABASE,X%,Y% results in the database name being translated from XMYDB to YMYDB. The DDL-ATTRIBUTE-CHANGE command will be generated as DDL-ATTRIBUTE-CHANGE(database,XMYDB, YMYDB,%,%) and results in every database string matching XMYDB changing to YMYDB in the generated DDL.

Recommendations

CAUTION:

DDL processing cannot be simulated. If you do not want DDL to be processed during a simulation (SIM (Y)), change DDL-ENABLE to N.

- When setting up a new clone operation, run the first time with DATA-MOVER PGM(NONE) to ensure the DDL is created and all the target objects are available for the copies. Then rerun the source job with PGM other than NONE, do the copies, and then run the target job. Once the source job is set up correctly, the DDL can be generated and the copies performed with one source job invocation.
- Specify all table spaces using a DATABASE or TABLESPACE object definition in LISTDEF and then specify ALWAYS-COPY-INDEXSPACES(Y).

For additional information about the DDL processing, refer to the topics “COPY” on page 473 and “DDL-ATTRIBUTE-CHANGE parameter values” on page 506.

Sample DDL processing scenarios

This topic contains several scenarios for processing DDL with DB2 Cloning Tool Table Space Cloning.

Generating and running target DDL when not all target objects exist

In this scenario, some target objects do not exist and copies of all the source objects are to be made. Settings for DB2 Cloning Tool Table Space Cloning to generate and run DDL for non-existent target objects and then copy the source objects to the target:

- PROCESS-TYPE(Y)
- PROCESS-DDL-DDN(ddname) - optional, to save the DDL used
- PGM(ADRDSSU)
- Add all OBJECT-TRANSLATE commands required to change names from source to target.
- Add any attributes that should be changed to the DDL-ATTRIBUTE-CHANGE command.

Changing the generated target DDL before running

In this scenario, you want to make changes to the DB2 Cloning Tool Table Space Cloning-generated DDL before running it. Generate the DDL in the first source job (using PGM(NONE) to prevent copies being made). Then run the DDL in the second source job.

Settings to generate DDL but not execute it, without copying the source objects to the target:

- PROCESS-TYPE(G)

- PROCESS-DDL-DDN(ddname)
- PGM(NONE)
- DEFVCAT(vcatvalue)
- COPY-IF-NO-DB2-TARGET-OBJECTS(Y)
- Add all OBJECT-TRANSLATE commands required to change names from source to target.
- Add any attributes that should be changed to the DDL-ATTRIBUTE-CHANGE command.

Edit the DDL generated in *ddname*. Then rerun DB2 Cloning Tool Table Space Cloning to execute the previously written and edited DDL. No copies will be performed. The DDL can also be submitted by SPUIFI or batch processing, outside of DB2 Cloning Tool Table Space Cloning. Once the target objects are complete, run DB2 Cloning Tool Table Space Cloning to make the copies.

- PROCESS-TYPE(X)
- PROCESS-DDL-DDN(ddname)
- PGM(NONE)

Use a source job with DLL-ENABLE(N) to copy objects from source to target.

Running generated target DDL when some target objects already exist

In this scenario, you have DDL for all of the target objects, however some target objects already exist and others do not. Settings for DB2 Cloning Tool Table Space Cloning to execute already-generated DDL and copy the source objects to the target:

- PROCESS-TYPE(X)
- PROCESS-DDL-DDN(ddname)
- IGNORE-CREATE-OBJECT-EXISTS(Y)
- PGM(NONE)

Use a source job with DLL-ENABLE(N) to copy objects from source to target.

Generating DDL for the source objects

In this scenario, you want a copy of the source DDL. Settings for DB2 Cloning Tool Table Space Cloning to generate the source DDL without executing it:

- PROCESS-TYPE(A)
- PROCESS-DDL-DDN(ddname)
- PGM(NONE)
- Do not specify OBJECT-TRANSLATE or DDL-ATTRIBUTE-CHANGE commands, and the DDL will reflect the source objects.

Generating DDL for the target objects

In this scenario, you want a copy of all target DDL, regardless of whether the object exists on the target. Settings for DB2 Cloning Tool Table Space Cloning to generate target DDL without executing it:

- PROCESS-TYPE(A)
- PROCESS-DDL-DDN(ddname)

- PGM(NONE)
- Add all OBJECT-TRANSLATE commands required to change names from source to target.
- Add any attributes that should be changed to the DDL-ATTRIBUTE-CHANGE command.
- Execute this source job using the old target subsystem as the LOCAL-SSID.

Considerations for using catalog prefetch to populate the object cache

This topic explains considerations when planning to use catalog prefetch.

Normally DB2 Cloning Tool Table Space Cloning reads the catalog using SELECTs with a WHERE clause. However, when indexes are not usable, one SELECT can take several seconds or longer. The CATALOG-PREFETCH command allows all the rows in SYSIBM.SYSTABLESPACE, SYSIBM.SYSTABLEPART, SYSIBM.SYSTABLES and SYSIBM.SYSCOLUMNS to be read without a WHERE clause. If any indexes are present in a LISTDEF or ALWAYS-COPY-INDEXES(Y) is specified, an attempt will be made to cache entries in SYSIBM.SYSINDEXES and SYSIBM.SYSINDEXPART.

DB2 Cloning Tool Table Space Cloning saves the prefetched objects based on the database and or table space objects already selected using LISTDEF statements or the databases in the prefetch database list.

CATALOG PREFETCH should be used whenever:

- V7-MIGRATED-OBJECTS-PRESENT(Y), or
- The DB2 catalog on the source or target subsystem is unable to use indexes due to do a poor cluster ratio.

If the source job is running with V7-MIGRATED-OBJECTS-PRESENT(N) and taking a long time to run, try REORGing the catalog. If REORGing the catalog is not an option, then using CATALOG-PREFETCH might be considered.

Using the source and/or target prefetch database list can reduce the fetch time for the source and/or target catalog. These commands allow all database objects to be saved in a single pass of the catalog tables being saved. When accessing the source catalog, and not using the prefetch database list, each new database results in another catalog access. When all source objects are in a single database and the object specification is DATABASE, the prefetch list provides no additional benefit in the source job. When using a TCP/IP server with KEEP-DATABASES-IN-DISCONNECT(Y), the target database list command can specify a list of databases to be used by multiple source jobs. Once the cache is populated in the TCP/IP server, no further catalog accesses for the cached tables are required during target catalog lookup.

Run the initial source job to populate the TCP/IP server cache with PGM(NONE) and specify all the databases using TARGET-PREFETCH-DATABASE-LIST. Then run each source copy job one at a time to access the cache in the TCP/IP server. Use REMOTE-CONNECT-TYPE(T) to use the TCP/IP server, even when the two subsystems are in the same LPAR.

When first setting up to copy a set of objects, run with different options and compare the elapsed times of the two jobs. Specify PGM(NONE) to prevent copies

and space stops and starts when running the comparison jobs. See CKZ61310I messages in the source job log data set for statistics on catalog prefetch performance.

CATALOG-PREFETCH has the following restrictions:

- This command is not available when running in DB2 Version 8 Compatibility Mode.
- There are restrictions on how LISTDEF statements can be coded. Refer to the “COPY command and keyword definitions” on page 476 topic for information about the LISTDEF requirements for the CATALOG-PREFETCH.
- Whenever the target objects do not exist, ENABLE-TARGET-PREFETCH(N) should be used. When the target objects exist, specify Y to use catalog prefetch for target objects.
- To cache source SYSTABLESPACE objects, you must specify a source database list. If there is no source database list, then table space caching is not performed. This does not affect caching of other catalog tables. Similarly, a target database list is required to cache target table spaces.
- Table space caching does not support masking characters in LISTDEF. This includes database and table space object specifications for both object types, TABLESPACES and INDEXSPACES. For example: INCLUDE TABLESPACES DATABASE MYDB% will result in no objects being found INCLUDE TABLESPACES DATABASE MYDB1 MYDB1 will be found (if it exists).
- Caching of only indexes is currently not supported.

When using CATALOG-PREFETCH, only a single subtask is used to access the catalog. A user-specified value greater than one will be changed in the source job to one. The original value will still be used for the target job.

Note that not all catalog tables are cached. Among those that are accessed without using the cache are SYSDATABASE, SYSKEYS, SYSFORIGNKEYS, and SYSSEQUENCES. Access to these is limited and should not substantially impact source job performance.

Considerations for caching indexes

There are several considerations when planning to cache indexes using catalog prefetch.

If any indexes are present in a LISTDEF or ALWAYS-COPY-INDEXES(Y) is specified, an attempt will be made to cache entries in SYSIBM.SYSINDEXES and SYSIBM.SYSINDEXPART. The object specification for index space type specifications in the LISTDEF must be database or table space, just as it is for table space type specifications. LISTDEF statements for index space type specifications must follow all table space LISTDEF type specification statements.

Indexes that are not in the databases selected with table space caching, and index spaces that are cached using a specification other than database or table space type specifications, will not be cached. Therefore, indexes in index only jobs will not be cached, as there are no table space type specification statements.

The simplest way to ensure all indexes are cached is to only specify table space type specifications using database or table space object definitions in LISTDEF, and then use ALWAYS-COPY-INDEXSPACES(Y). If the COPY attribute is needed, then individual index statements must be specified with the COPY attribute added.

Note that when using ALWAYS-COPY-INDEXES(Y), the internally created object spec is the same as it is for the corresponding table space type specification.

To include all index spaces in a database using COPY NO (the default), specify the LISTDEF and command as follows:

LISTDEF statements:

```
INCLUDE TABLESPACES DATABASE yourdb1 ALL
INCLUDE TABLESPACES DATABASE yourdb2 ALL
INCLUDE TABLESPACES DATABASE yourdb3 ALL
```

COPY command:

```
ALWAYS-COPY-INDEXSPACES(Y)
```

To include all index spaces in a database using COPY YES (not the default), specify the LISTDEF as follows:

```
INCLUDE TABLESPACES DATABASE yourdb1 ALL
INCLUDE TABLESPACES DATABASE yourdb2 ALL
INCLUDE TABLESPACES DATABASE yourdb3 ALL
INCLUDE INDEXSPACES COPY YES DATABASE yourdb1 ALL
INCLUDE INDEXSPACES COPY YES DATABASE yourdb2 ALL
INCLUDE INDEXSPACES COPY YES DATABASE yourdb3 ALL
```

Using a local TCP/IP server to populate the target cache for multiple source copy jobs

This topic describes the procedure to use a local TCP/IP server to populate the target cache for multiple source copy jobs.

About this task

There are three copy jobs to be run.

Procedure

1. Specify SET KEEP-DATABASES-ON-DISCONNECT(Y) in the TCP/IP job parms.
2. Start the TCP/IP server on the LPAR used by the target subsystem.
3. In a pre-copy source job, specify:
 - a. PGM(NONE)
 - b. SET REMOTE-CONNECT-TYPE(T)
 - c. Add parms for catalog prefetch and use TARGET-PREFETCH -DATABASE-LIST to include the databases for all three jobs.
4. Run the job to populate the TCP server cache with all the target objects.
5. In all source copy jobs, specify:
 - a. SET REMOTE-CONNECT-TYPE(T)
 - b. Add parms for catalog prefetch.
 - c. Use the same TARGET-PREFETCH -DATABASE-LIST that was used in step 4.
 - d. PGM(ADRSSU) to do the copies.
6. Run source copy job 1.
7. Run source copy job 2.
8. Run source copy job 3.

Results

When steps 6, 7 and 8 are run, there will no DASD access for the three catalog tables that are cached to obtain target job information.

Return code choices

The CKZINI member in the product PARMLIB allows you to choose the seriousness of a return code. Scenarios discussed in this document assume that the conditional execution of subsequent steps adheres to the convention that return code 0 means successful, 4 means warning, and 8 means an error..

The following control parameters are in the product PARMLIB member CKZINI:

```
MAX_RC = 0           /* stop job when return code is greater than MAX_RC */
MAX_COPY_RC = 0     /* stop job when return code is greater than MAX_COPY_RC */
                   /* occurs when copying data */
```

These control parameters can be overridden by using the MAX-RC or the MAX-COPY-RC keywords of the SET command.

Plans and packages

Plans and packages must be bound on both the source and target DB2 systems before using DB2 Cloning Tool Table Space Cloning.

If you need to bind the plans, refer to Chapter 4, “Customizing DB2 Cloning Tool,” on page 51.

Dropped and altered tables

If any of the following is true, you might need to run a REORG on the source and/or target table spaces before copying them.

- One or more tables were dropped from the table space.
- If an ALTER TABLE ADD COLUMN was executed on the table space.
- If objects have been migrated from DB2 V7 and have had column(s) added using ALTER TABLE ADD COLUMN, note that the status AREO* may not be displayed. You must REORG the table spaces before using DB2 Cloning Tool Table Space Cloning to copy them. Failure to perform this REORG may result in the target data being inaccessible.

Copying versioned objects

Use the procedure in this topic when it is possible one or more versioning ALTER commands have been done on any table space, table or index space in a DB2 Cloning Tool Table Space Cloning LISTDEF.

About this task

When an object is altered, DB2 places versioning information in the catalog, DB2 directory and the page set. REORG and REPAIR can correct the DB2 catalog and directory, but not the page sets with table OBIDs. However, the DB2 Cloning Tool Table Space Cloning source job can change the source table OBIDs to the target table OBIDs.

In addition, you can use the `WARN-ON-VERSIONS` subcommand of the `COPY` command to request an informational or warning message for all table spaces and index spaces with a non-zero version count. These messages may indicate that a `REORG` and `REPAIR` utility must be run along with DB2 Cloning Tool Table Space Cloning processing.

Run the following procedure to ensure that the target object(s) copied is (are) accessible:

Procedure

1. Run `REORG` on all source and target table spaces with versioning.
2. Run `REBUILD` on all source and target indexes with versioning.
3. Run the DB2 Cloning Tool Table Space Cloning source job.
4. Run the DB2 Cloning Tool Table Space Cloning target job.
5. Run `REPAIR VERSIONS` for each table space on the target that has a mismatch between the pages in the page set and the target DB2 directory. If in doubt, run `REPAIR` once to ensure page set and directory synchronization.

What to do next

Once these steps have been done, change the `WARN-ON-VERSIONS` command back to `N` to turn off warning messages that do not apply.

The previous procedure does not reset the version to zero, so DB2 Cloning Tool Table Space Cloning continues to output an informational message for each object with a non-zero version number in the catalog. To reset the version to zero, resubmit `CREATE DDL`.

Once all versioned objects in a source job have had `REORG` and `REPAIR` run, no further utility processing is required until another `ALTER` is done to one or more objects. The DB2 Cloning Tool Table Space Cloning target job continues to translate all table versioning OBIDs in each subsequent run with no intervention required.

Note that DB2 Cloning Tool Table Space Cloning does not detect whether the version has been corrected using `REPAIR` and the messages associated with any versioned object will continue to print until the version number is reset to zero using `CREATE DDL`.

Chapter 13. DB2 Cloning Tool Table Space Cloning function overview

This topic offers a high level overview of the functions and jobs for using DB2 Cloning Tool Table Space Cloning.

DB2 Cloning Tool Table Space Cloning is comprised of the source job, the target job, and an optional TCP/IP server job.

Source job overview

The DB2 Cloning Tool Table Space Cloning source job identifies all DB2 table spaces and index spaces to be copied to the target DB2 subsystem.

The source job identifies the spaces to be copied by:

- Selecting the table spaces and index spaces to be cloned via a LISTDEF facility, similar to the IBM LISTDEF
- Collecting information from both the source and target DB2 catalogs
- Deriving the names of the VSAM data sets for the source DB2 table spaces and index spaces
- Deriving the names of the VSAM data sets for the target DB2 table spaces and index spaces
- Issuing STOP commands for the target table spaces and index spaces (if requested by parms) and the source table spaces and index spaces
- Issuing the fast replication copy of the VSAM objects for FlashCopy or SnapShot using DFSMSdss or TimeFinder/Clone
- Generating the parameters and SQL that the target job uses to make the copied table spaces and indexes accessible at the target
- Writing the generated parameters and SQL to a sequential data set or PDS member which will be used as input for the target job that executes on the target DB2 z/OS system

The source job consists of 4 logical phases:

1. Init Phase, described in “Source job - Init phase I” on page 170.
2. Discovery Phase, described in “Source job - Discovery phase II” on page 171.
3. Eligibility Phase, described in “Source job - Eligibility phase III” on page 171.
4. Copy Phase, described in “Source job - Copy phase IV” on page 172.

Target job overview

The DB2 Cloning Tool Table Space Cloning target job makes the data accessible to the target DB2.

The target job makes the data accessible by:

- Translating the object IDs from the source table spaces and index spaces to those of the target table spaces and index spaces
- Resetting the LOGRBA on the target
- Optionally, starting the target table spaces and index spaces for RW (read/write) access

A target job repository can be defined and used to keep track of target jobs and the data sets that are processed by the target jobs. This repository allows the failed target job to be restarted, skipping any successfully processed target data sets. The USE-RUNTIME-REPOSITORY parameter of the SET command determines how the target job is processed. If set to N, the target job starts over. If set to Y, the target job skips all data sets that have been processed or were being processed when the target job ended.

The target job has one logical phase, the Resolution Phase.

TCP/IP job overview (optional)

In the absence of a connection via CAF (DB2 Call Attach Facility), the DB2 Cloning Tool Table Space Cloning TCP/IP server job facilitates the communication of the DB2 Cloning Tool Table Space Cloning source job with a remote DB2 target subsystem. It is also used to issue DB2 commands from the source z/OS system to the target z/OS system when a DDF connection to the target subsystem is in use.

Report job overview (optional)

The report job is a separate job that simply outputs the data in the target job runtime repository and ends.

The source job

The source job identifies all DB2 table spaces and index spaces to be copied to the target DB2 subsystem using the DB2 Cloning Tool Table Space Cloning LISTDEF command, similar to IBM LISTDEF.

The default name of the source job in DB2 Cloning Tool Table Space Cloning is CKZxSRC, where x is a number 1-4 to match the setup scenarios in Chapter 14, "Setup procedures for copy by data set with FlashCopy, SnapShot, or TimeFinder/Clone," on page 179 and Chapter 15, "Set up procedures for copy by data set for all other methodologies," on page 199. The name for the target job is CKZTRG. Sample jobs can be found in the product JCL library.

Note: The source job should be scheduled when no utilities are running and there is no read/write activity against the table spaces and index spaces being copied.

The source job consists of four phases, documented in topics that follow.

Source job - Init phase I

The source job reads all DB2 Cloning Tool Table Space Cloning settings found in the product PARMLIB member CKZINI and in the CKZIN DD statement.

The CKZIN DD statement identifies the following source and target items:

- Source (local) and target DB2 subsystem names
- DDs passed to ADRDSSU or the EMC API for data set allocations
- DB2 location name for DDF (optional)
- Security ID and password for DDF and IP address for TCP/IP (optional)
- A high level qualifier for the target VSAM data sets (if the DB2 table spaces and index spaces do not exist in the target DB2 catalog)
- Source/target creator ID pairs. A creator ID is the first qualifier of a DB2 table

- Copy options

Source job - Discovery phase II

The source job reads the DB2 Cloning Tool Table Space Cloning LISTDEF DD statement (CKZLSTDF) to determine the table spaces and index spaces to collect information from, in both the DB2 source and DB2 target catalogs. This phase determines what SQL queries are used to extract information from the source and target DB2 catalogs. No DB2 table spaces and index spaces are stopped during LISTDEF processing.

The Discovery Phase performs these tasks:

- Determines the source DB2 subsystem name.
- Determines all candidate source objects (databases, table spaces, tables, index spaces and indexes) to be used to determine data sets to be copied to the target DB2 subsystem based on the DB2 Cloning Tool Table Space Cloning LISTDEF control statements. The user may optionally specify partitions of a partitioned table space or index space, RI or ALL (LOB specification).
- Eliminates any duplicate table spaces and index spaces specified in the LISTDEF command. Using the LISTDEF example that follows, all table spaces in database XXXXX are included in the copy except for those that begin with 'PRODAB. '. However, the third parameter is in error. The intent may have been to include one or more of the excluded table spaces. Instead it will override the exclude. Table spaces beginning with 'PRODA' will be duplicated. DB2 Cloning Tool Table Space Cloning will eliminate the duplicates caused by statement 1 and 3.


```
INCLUDE TABLESPACES DATABASE XXXXX
EXCLUDE TABLESPACES TABLESPACE XXXXX.PRODAB%
INCLUDE TABLESPACES TABLESPACE XXXXX.PRODA%
```
- Selects the list of candidate DB2 table spaces and index spaces from the source DB2 catalog and records:

Qualifiers (VCATs, creator IDs)
 Object types (tables, indexes etc.)
 Object names
 Object attributes
 Column attributes
 Object IDs
 and others

- Constructs the names of the VSAM data sets associated with the DB2 table spaces and index spaces.
- If LOB tables are present, DB2 Cloning Tool Table Space Cloning verifies that both the base tables and auxiliary table spaces are present for the copy and warns if one of them is missing.

Source job - Eligibility phase III

This phase confirms that a compatible target table space or index space exists on the target DB2 subsystem. Comparison of some of the attributes between the source and target table space or index space are made to determine if the data set can be copied from the source to the target.

For example, DSSIZE, page size (buffer pool), parted, CCSID, etc., are compared. Any attributes that may cause a conflict between the content or representation of the data in the source and target VSAM objects are compared.

Target analysis

- Uses TARGET_DB2(SSID(XXXX)) from the DB2 Cloning Tool Table Space Cloning COPY command to determine the target DB2 subsystem
- Uses CAF, DDF or TCP/IP to connect to the target DB2 subsystem
- Pairs the proper target creator ID with each source creator ID
- Verifies the existence of table spaces and index spaces on the target that match the candidate table spaces and index spaces on the source. If matching table spaces and index spaces are found, DB2 Cloning Tool Table Space Cloning compares:
 - Qualifiers
 - Object names
 - Object attributes
 - Column attributes
 - Object IDs
 - Buffer pools
 - CCSID
 - and others
- If no matching table space or index space is found and the COPY command keyword COPY-IF-NO-DB2-TARGET-OBJECTS(Y) is in effect, DB2 Cloning Tool Table Space Cloning:
 - Constructs a VSAM data set name based on the information for the source DB2 table space or index space
 - Uses the DEFVCAT qualifier as the HLQ for the VSAM data set name
 - Records its object IDs as zeroes
- If no matching table space or index space is found and the COPY command keyword COPY-IF-NO-DB2-TARGET-OBJECTS(N) is in effect, DB2 Cloning Tool Table Space Cloning sends a warning (CC 4) and records the missing table space or index space name(s) in CKZPRINT
- If the source DB2 table space or index space is a partitioned table space or index space, DB2 Cloning Tool Table Space Cloning extracts DB2 catalog information about all of the target partitions in a single pass.
- If one or more target table spaces or index spaces do not exist in the catalog and you choose to copy the VSAM data set by specifying the COPY-IF-NO-DB2-TARGET-OBJECTS(Y) keyword, then DB2 Cloning Tool Table Space Cloning will not be able to extract the target object IDs for those data sets. At that time the object IDs will not exist. After the DB2 table spaces and index spaces are created by you, the source job may be re-run with the COPY command keyword DATA-MOVER(PGM(NONE)). The source job will update the SYNCDB2 data set with the missing object IDs.

Source job - Copy phase IV

This phase determines if FlashCopy, SnapShot, or TimeFinder/Clone will be used to replicate the table spaces and index spaces, or if another copy mechanism will be used and prepares the parameters and SQL necessary to make the VSAM objects accessible at the target DB2.

DB2 Cloning Tool Table Space Cloning will invoke FlashCopy or SnapShot (if available) using DFSMSdss, program ADRDSSU, or uses the EMC API to invoke EMC TimeFinder/Clone, or you can use any other copy mechanism to replicate or copy the list of candidate data sets identified in the Eligibility Phase.

A source and target VSAM object pair is passed to the COPY Phase if:

- The attributes of the DB2 table spaces and index spaces associated with the source and target data set pairs are compatible between the source and the target subsystems
- If the object is a LOB, both the base and auxiliary table spaces must be included in the copy selection
- All table or index column order, column attributes and lengths are identical between the source and target table spaces and index spaces
- The target table spaces and index spaces exist or the keyword is COPY-IF-NO-DB2-TARGET-OBJECTS(Y)
- The target data set does exist and the keyword is REPLACE-TARGET-DSN(Y), or the target data set does not exist and the keyword is REPLACE-TARGET-DSN(N)
- The spaces to be copied are not in certain restricted or advisory states (if requested in the SET command)

Copy steps

1. Copy source table spaces and index spaces to the target DB2.
 - Using FlashCopy, SnapShot, or TimeFinder/Clone: Unless the optional FUZZY-COPY is specified, DB2 Cloning Tool Table Space Cloning deallocates the source VSAM objects from the DB2 Database Manager address space prior to the DB2 Cloning Tool Table Space Cloning invocation of DFSMSdss, program ADRDSSU, to execute FlashCopy, SnapShot, or prior to invoking the EMC API for TimeFinder/Clone.
 - Using all other copy methodologies: The user is responsible for the deallocation of the source VSAM objects, and the copy methodology.
2. DB2 Cloning Tool Table Space Cloning restarts the source table spaces and index spaces after the copy is successful.
3. DB2 Cloning Tool Table Space Cloning creates control parameters in a sequential data set or a user-specified member of a PDS to:
 - Translate object IDs from the source table spaces and index spaces to the target table spaces and index spaces to make them accessible
 - Optionally, reset the LOGRBA for the target table spaces and index spaces
 - Optionally, start the target DB2 table spaces and index spaces after they are made accessible
 - Optionally, apply data masking to target tables, if requested
4. DB2 Cloning Tool Table Space Cloning may create SQL in a sequential data set or a user-specified member of a PDS to update any identity column information on the target.
5. DB2 Cloning Tool Table Space Cloning passes the RESET-LOGRBA parameter to the target job. This option resets the RBAs for the target VSAM objects. The LOGRBA will always be reset if there are OBID changes to be made. The level ID is always reset. Resetting the level IDs prevents DB2 from rejecting the data set as too old or unknown.

The target job

The DB2 Cloning Tool Table Space Cloning target job must be submitted on the same z/OS system as the target DB2 subsystem by the user or job scheduler. This job resolves any outstanding issues with the target DB2 table spaces and index spaces to make them accessible.

Target job steps

Execute the commands and SQL passed from the DB2 Cloning Tool Table Space Cloning source job to:

- Use SQL to update any necessary ID column data
- Execute object ID translation for target DB2 table spaces and index spaces
- Optionally, reset the LOGRBA
- Optionally, apply DB2 logs to target tables, if requested
- Optionally, start the DB2 table spaces and index spaces
- Optionally, apply data masking to target tables, if requested

Once this job ends, the cloned table spaces and index spaces are accessible to DB2.

The TCP/IP server job (optional)

The optional DB2 Cloning Tool Table Space Cloning TCP/IP server job facilitates communication between the DB2 Cloning Tool Table Space Cloning source job and a target DB2 subsystem on a different LPAR. This is only necessary when a DDF connection or a TCP/IP connection is used for the target subsystem.

The DB2 Cloning Tool Table Space Cloning source job can act as a TCP/IP client job. If a CAF or DDF connection cannot be established between the source and target DB2 subsystems, then the source job automatically invokes TCP/IP to route requests to the target DB2 via the TCP/IP server job. The TCP/IP server job, CKZTCPS, is submitted by the user and executes on the target z/OS system. CKZTCPS must be started before submitting the source job that requires the use of DB2 Cloning Tool Table Space Cloning TCP/IP client. TCP/IP must be active on the same LPAR as the source and target DB2 subsystems.

TCP/IP server job steps

The TCP/IP server job has only one step and executes the same program as the source job.

The SET command keyword TCP-SERVER-JOB(Y) shifts the behavior of the job to that of communications facilitator.

The DB2 Cloning Tool Table Space Cloning source job will attempt to connect to the target DB2 in the following order:

1. CAF
2. DDF
3. TCP/IP

If the REMOTE-CONNECT-TYPE keyword is included on the SET command, DB2 Cloning Tool Table Space Cloning attempts to connect using the specified connection type (CAF, DDF, or TCP/IP). If the specified connection type fails, DB2 Cloning Tool Table Space Cloning does not attempt to connect via the other two connection types.

The TCP/IP server job, CKZTCPS, must execute on the same z/OS system as the target DB2. The control parameters for CKZIN DD of the server job are:

```
SET LOCAL-SSID(trss) TCP-SERVER-JOB(Y)
```

where:

trss is the target DB2 subsystem

keyword TCP-SERVER-JOB(Y) causes standard DB2 Cloning Tool Table Space Cloning JCL using PGM= CKZ00500 to become a server job.

The TCP/IP server job uses port 5099 as the default. The same port must be used by the client job. The parameters used by both the server job and the client job are set using tokens in the TCPIP_OPTIONS section of *hlq1.SCKZPARM(CKZINI)*, as follows:

```
:TCPIP_OPTIONS
  TCPIP_STC_NAME      = TCPIP          /* LOCAL TCPIP STC NAME
  TCPIP_SERVER_PORT  = 5099           /* TCPIP SERVER PORT #
```

where:

TCPIP is the name of the local TCP/IP address space on the source side

5099 is the port on which the TCP/IP server is listening on the target side

hlq1 is the data set high level qualifier used for the DB2 Cloning Tool Table Space Cloning PARMLIB member. However, it is not required that the client and server jobs use the same PARMLIB member. If, for example, the name of the TCP/IP address spaces differ on the source and target z/OS systems, different PARMLIB members must be used. You must make certain that the port names and numbers in both the source and target PARMLIB members still match. Or, you can set the port names and numbers using the TCPIP-SERVER-PORT and TCPIP-STC-NAME parameters of the SET commands in the source or TCPIP jobs.

The IP address of the target z/OS system is required. It is located in the COPY command. Support is available for both IPv4 and IPv6. To use IPv6, specify IP-VERSION6(Y) on the TCP/IP server SET command and specify an IPv6 address (using : for the field separator) in the client COPY command.

CKZTCPS requires an initiator and must be accessed by client jobs in series. That is, only one client can be connected to a specific server at any one time. To run another server on the same z/OS system, submit another job with a different port number and/or TCP/IP started task name.

If you are running multiple source jobs concurrently, you should have a different TCP/IP server job running for each source job. Each TCP/IP server and source job must use a unique TCP/IP port number. The same server job can be used if the source jobs are run in series.

A separate TCP/IP server is required for each target subsystem that is accessed concurrently on a particular LPAR. Each TCP/IP server must have its own port for the client to connect to.

Stopping CKZTCPS

Once submitted, CKZTCPS will execute until canceled or the time limit is exceeded. It may be canceled via the standard JES2 cancel command:

```
C XXXXXXX
```

The report job (optional)

The optional DB2 Cloning Tool Table Space Cloning report job produces a report of the contents of the target job runtime repository.

If you plan to run a report job, ensure that the repository data sets are allocated before the source and target jobs are run. Data set allocations are contained in the SCKZJCL library member CKZRREP.

The report job output can be two different widths. The short form is RECFM=FBA, LRECL=133. The long form is RECFM=FBA, LRECL=201. Both report formats can be obtained in the same report, as follows:

```
//CKZDREPL DD DISP=SHR,DSN=&dsnlong(&mbr)
//CKZDREPS DD DISP=SHR,DSN=&dsnshor(&mbr)
```

A sample report job is contained in the SCKZJCL library member CKZREPJB.

Report layout

These columns apply to both the short and long form of the report.

Table 26. Columns on both short and long form of the report

Column	Description
type	This column contains RRJB.
src job date	The date the source job was started.
src job time	The approximate time the source job started.
src job name	The source job name.
src job number	The source job JES number.
nr	The Iteration of the target job: 01 if only run once. There is a row in the report each time the target job is run or rerun.
trg system	The target z/OS system id.
trg job date	The date of the target job was started.
trg job time	The approximate time the target job was started.
trg job name	The target job name.
trg job number	The source job JES number.
src SSID	The DB2 source subsystem.
trg SSID	The DB2 target subsystem.
trg job RC	The target job return code.
trg job RS	The target job reason code.

Table 27. Columns on the short form of the report

Column	Description
type	This column contains RRDS.
trg job date	The date the target job was started.
trg job time	The approximate time the target data set processing started
trg data set name	The name of the target data set.
data set RC	The two-digit data set return code.
elapsed time in seconds	The elapsed time to process the data set in seconds.

Table 27. Columns on the short form of the report (continued)

Column	Description
space type	The space type, either tablespace or indexspace.
clone	This column is blank if the data set is not a clone data set. If the data set is a clone, the column contains B if a clone base and C if a clone.
# log pages changed	The number of pages that had one or more log records applied.
# data masking pages changed	The number of pages that had one or more data masking changes applied.
# VSAM reads	The total number of VSAM reads, including zero pages.
# VSAM writes	The total number of changed pages.

Table 28. Columns on the long form of the report

Column	Description
type	This column contains RRDS.
trg job date	The date the target job was started.
trg job time	The approximate time the target data set processing started.
src data set name	The name of the source data set.
trg data set name	The name of the target data set.
data set RC	The two-digit data set return code.
pages	The total number of pages in this data set.
elapsed time in seconds	The elapsed time to process the data set, in seconds.
space type	The space type, either tablespace or indexspace.
clone	This column is blank if the data set is not a clone data set. If the data set is a clone, the column contains B if a clone base and C if a clone.
# log pages changed	The number of pages that had one or more log records applied.
# data masking pages changed	The number of pages that had one or more data masking changes applied.
# VSAM reads	The total number of VSAM reads, including zero pages.
# VSAM writes	The total number of changed pages.
IO err	This column contains Y if an I/O error occurred.
VSAM err	This column contains Y if a VSAM error occurred.

CKZFIX job

The optional DB2 Cloning Tool Table Space Cloning fix job, CKZFIX, facilitates dealing with missing target DB2 table spaces and index spaces.

The CKZFIX job uses parameters automatically generated by the source job to make the data for the missing table spaces and index spaces available to DB2. Sample JCL for this job is in the product JCL library. You must create the missing DB2 table spaces and index spaces either manually or via a tool and then run the fix job. CKZFIX must execute on the target z/OS system.

CKZFIX job steps

1. Stop: Stops the newly created (formerly missing) DB2 table spaces and index spaces. This deallocates them from DB2. This requires a resubmittal of the source job after the table spaces and index spaces are created.
2. Delete/ rename: Deletes the VSAM data set created by DB2 and renames the replicated VSAM data set to take the place of the deleted data set.

CKZSTPT, CKZSTPS and CKZSTRS jobs

The optional CKZSTPT, CKZSTPS and CKZSTRS jobs facilitate in replicating DB2 data sets outside of DB2 Cloning Tool Table Space Cloning. They stop the target DB2 table spaces and index spaces, stop the source DB2 table spaces and index spaces and restart the source DB2 table spaces and index spaces.

These jobs are user submitted. At the user's request, DB2 Cloning Tool Table Space Cloning will generate parameter for these jobs. The JCL for these jobs is in the product JCL library.

- CKZSTPT has only one step. This job submits DB2 commands to stop the target DB2 table spaces and index spaces and to deallocate the target VSAM data sets. It must execute on the target z/OS system.
- CKZSTPS has only one step. This job submits DB2 commands to stop the source DB2 table spaces and index spaces and to deallocate the source VSAM data sets. It must execute on the source z/OS system.
- CKZSTRS has only one step. This job submits DB2 commands to start the source DB2 table spaces and index spaces after they have been copied to the target.

Chapter 14. Setup procedures for copy by data set with FlashCopy, SnapShot, or TimeFinder/Clone

This topic contains DB2 Cloning Tool Table Space Cloning setup procedures for you to use if data set copies are to be created with FlashCopy or SnapShot via DFSMSdss or TimeFinder/Clone.

Setup procedures for scenarios using methodologies other than FlashCopy, SnapShot, or TimeFinder/Clone are documented in Chapter 15, "Set up procedures for copy by data set for all other methodologies," on page 199.

Setup for scenario 1: FlashCopy, SnapShot, or TimeFinder/Clone - All target table spaces and index spaces already exist in DB2 catalog

Use the procedure in the following table when FlashCopy, SnapShot, or TimeFinder/Clone is used to copy source table spaces and index spaces to the target DB2 subsystem when the target table spaces and index spaces already exist in the DB2 catalog.

Table 29. DB2 Cloning Tool Table Space Cloning - Setting up for using FlashCopy, SnapShot, or TimeFinder/Clone when all target objects already exist

FlashCopy, SnapShot, or TimeFinder/Clone Setup Steps
"Step 1: Create target DB2 table spaces and index spaces in the target DB2 catalog."
"Step 2: Set up and execute source job (CKZ1SRC)" on page 180.
"Step 3: Set up and execute target job (CKZTRG)" on page 184.

Step 1: Create target DB2 table spaces and index spaces in the target DB2 catalog

The first step when copying by data set using FlashCopy, SnapShot, or TimeFinder/Clone is to create the target objects in the target DB2 catalog.

DB2 associates DB2 table spaces and index spaces with specific VSAM data set names. For example:

```
VCAT.DSNDBD.DBTEST1.TESTTBSP.I0001.A001
```

where:

Node 1: VCAT - the data set high level qualifier

Node 2: DSNDBD is the data portion. DSNDBC is the cluster portion

Node 3: DBTEST1 - the database name in the DB2 catalog

Node 4: TESTTBSP - the table space or index space name in the DB2 catalog

Node 5: Either I0001 or J0001 (a switch indicator)

Node 6: A001 to E094, a partition number or DB2 extent number

The correctly named DB2 table spaces and index spaces must exist in the target DB2 catalog for DB2 to recognize and use the VSAM data sets cloned to the target by DB2 Cloning Tool Table Space Cloning.

When DB2 creates VSAM objects, it puts internal object identifiers inside of them. Those internal object identifiers are probably not the same on the target as the internal object identifiers in the VSAM data sets cloned from the source. In particular, the database ID will almost never be the same; however, the other IDs are relative to the database and will frequently be the same. DB2 Cloning Tool Table Space Cloning will have to translate the internal identifiers from the source to those of the target to make the cloned VSAM data sets usable by the target DB2.

Create as many of the following as are required by your application:

- Create the target stogroup(s)
- Create the target database(s)
- Create the target table space(s)
- Create the target table(s)
- Create the target index space(s)
- Create any target primary keys
- Create any target foreign keys
- Create any target alias(es), view(s), synonym(s), constraint(s), trigger(s), stored procedure(s), etc.

Note: DB2 Cloning Tool Table Space Cloning can be used to find target table spaces and index spaces that do not exist. Submit the source job with PGM(NONE) and copy parameter COPY-IF-NO-DB2-TARGET-OBJECTS(N). DB2 Cloning Tool Table Space Cloning will issue a warning message each time a source table space or index space cannot be paired with a target table space or index space.

Step 2: Set up and execute source job (CKZ1SRC)

The source job is comprised of several commands and phases to control the selection of the DB2 table spaces and index spaces on the source subsystem and to make them accessible on the target subsystem.

About this task

An example of the source JCL for this scenario can be found in the product JCL library member CKZ1SRC. A detailed example is provided for illustration after the procedure steps. Each step contains a reference to the example.

Procedure

1. Identify the source table spaces and index spaces to DB2 Cloning Tool Table Space Cloning using the DB2 Cloning Tool Table Space Cloning LISTDEF command.
 - The LISTDEF control statements can be either be entered into an existing PDS member (RECFM=FB,LRECL=80) and read as input by using the CKZLSTDF DD statement. See line #6 in the source job example and the associated note 6.
 - Or, the LISTDEF control statements can be read in-stream by using CKZLSTDF DD. See line #6 in the source job example and the associated note 6.
2. Specify the output data set where DB2 Cloning Tool Table Space Cloning will build the input parameters for the target job. See line #7 in the source job example and the associated note 7.

3. Identify the source DB2 subsystem in the SET command, keyword LOCAL-SSID(n), where n is the name of your source DB2 subsystem. See line #9 in the source job example and the associated note 9.
4. If input and/or output DDs must be passed to ADRDSSU, use the HLQDDDF command to specify those DDs. Identify the HLQ (high level qualifier) and a DD name or names using keyword HLQNAME(n) and DDNAME(dd1...,ddn), where n is your high level qualifier and dd1...ddn is one or more DDs. See line #10 in the source job example and the associated note 10.
5. Identify the target DB2 subsystem information and copy method in the COPY command.
 - Identify the target DB2 subsystem to CKZIN in the COPY command, keyword TARGET-DB2(SSID(n) where n is the name of your target DB2 subsystem. See line #11 in the source job example and the associated note 11.
 - If using data set level FlashCopy or SnapShot to copy table spaces and index spaces, and your environment is set up to support data set level replication, use COPY command keyword, DATA-MOVER(PGM(ADRDSSU)) which invokes DFSMSdss program ADRDSSU to execute either FlashCopy or SnapShot to do the data set level copy operation. If FlashCopy is not available, ADRDSSU will substitute a normal copy operation, unless REQUIRED is specified. See line #11 in the source job example and the associated note 11. If using EMC TimeFinder/Clone, specify COPY command keyword DATA-MOVER(PGM(EMCAPI)).

Results

CAUTION:

If COPY command keyword FUZZY-COPY(Y) is specified, then DB2 Cloning Tool Table Space Cloning will not stop the source table spaces and index spaces. However, we do not recommend this, because if the table spaces and index spaces are in RW status, there are data integrity issues.

Source job example

For completeness and to illustrate where JCL parameters and DB2 Cloning Tool Table Space Cloning control statements must match, the following JCL includes sample DB2 Cloning Tool Table Space Cloning control statements.

The source job JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

//jobcard
/*JOBPARM S=srcsys
//*
//*****
//*      COPYRIGHT 2009 MAINSTAR SOFTWARE CORPORATION      *
//*                               ALL RIGHTS RESERVED      *
//*****
//*
//* Scenario 1 - all spaces to be copied exist on the target and
//*                CKZ controls the copy process
//*
//* variables to be filled in ...
//*
//* jobcard - job card
//* srcsys - name of source system
//* hlq1 - CKZ LOADLIB and PARMLIB high level qualifier
//* hlq2 - SYSMDUMP high level qualifier
//* dsnvrl - DB2 high level qualifier

```

```

/**      parmbr  - CKZ PARMLIB member
/**      dumpsc  - SYSDUMP STORCLAS if needed
/**      dumpmc  - SYSDUMP MGMTCLAS if needed
/**      hlq3    - CKZ users high level qualifier
/**      lstdmbr - LISTDEF member name
/**      trss    - target DB2 subsystem
/**      dddd    - DD DISP value if needed (must enable)
/**      uuuu    - DD UNIT value if needed (must enable)
/**      vvvvvv  - DD VOLUME value if needed (must enable)
/**      srss    - source DB2 subsystem
/**      defsqlid - default SQLID applied to LISTDEF objects if needed
/**      tloc    - target subsystem if using DDF (delete if not needed)
/**      ipaddr  - target IP if using TCPIP (delete if not needed)
/**      uid     - user id if using DDF (delete if not needed)
/**      password - password if using DDF (delete if not needed)
/**      ocs     - object creator on source DB2 (delete if not needed)
/**      oct     - object creator on target DB2 (delete if not needed)
/**
/**
/**      Initiate DFSMSdss to EXECUTE FLASHCOPY or SNAPSHOT
/**
1 //STEP1  EXEC PGM=CKZ00500,REGION=0M
2 //STEPLIB DD  DISP=SHR,DSN=hlq1.SCKZLOAD
//          DD  DISP=SHR,DSN=dsnvr1.SDSNEXIT
//          DD  DISP=SHR,DSN=dsnvr1.SDSNLOAD
/**
/**
/**      CKZINI - Reads the CKZ parmlib to get default settings
/**
3 //CKZINI  DD  DISP=SHR,DSN=hlq1.SCKZPARAM(parmbr)
/**
/**
/**      CKZPRINT - primary message output, required
/**
4 //CKZPRINT DD  SYSOUT=*
/**
/**
/**      CKZLOG - detailed message output, optional
/**
5 //CKZLOG  DD  SYSOUT=*
/**
/**
/**      dump DDs
/**
/**      *SYSUDUMP DD  SYSOUT=*          Note: SYSDUMP preferred
/**      *SYSDUMP DD  DSN=hlq2.SYSDUMP,DISP=(NEW,CATLG),
/**              STORCLAS=dumpsc,MGMTCLAS=dumpmc,
/**              SPACE=(CYL,(50,30),RLSE),
/**              DCB=(LRECL=4160,BLKSIZE=4160,RECFM=FB),UNIT=SYSDA
/**      *ABNLIGNR DD  DUMMY          do not remove if using ABENDAID
/**
/**
/**      CKZLSTDF - CKZ uses LISTDEF like commands with standard IBM syntax
/**      to select the source spaces to be copied to the target.
/**
6 //CKZLSTDF DD  DISP=SHR,DSN=hlq3.LISTDEF(lstdmbr)
/**
/**
/**      CKZSYNC - output commands that will be used by the target
/**      job to make the VSAM object(s) accessible on the target
/**      DB2 subsystem.
/**
7 //CKZSYNC  DD  DISP=OLD,DSN=hlq3.SYNCDB2(lstdmbr)
/**
/**
/**      CKZSQL - output ALTER TABLE SQL needed to synchronize
/**      identity column values between target and source.

```

```

8  /**
   //CKZSQL DD DISP=OLD,DSN=hlq3.SQLOUT(1stdmbr)
   /**
   /**
   /** DSS output DD if required, see HLQDDDF below
   /**
   /***MYOUTDD DD DISP=dddd,UNIT=uuuu,VOL=SER=(vvvvvv)
   /**
   /*******
   /** sample control statements
   /*******
   /**
   //CKZIN DD *
   /**
   /**
   /** Source DB2 subsystem name and default SQLID if needed
   /**
   /**
9  SET LOCAL-SSID(srss) DEFAULT-SQLID(defsqlid)
   /**
   /**
   /** CKZ sends this DD name to ADRDSSU as output (enable if needed)
   /**
   /**
10 /** HLQDDDF HLQNAME(trghlq) DDNAME(MYOUTDD)
   /**
   /**
   /** The COPY command invokes DFSMSdss program, ADRDSSU, to
   /** execute FlashCopy or SnapShot and prepares the object ID
   /** translation parameters for the target job. Or, if COPY
   /** keyword DATA-MOVER(PGM(NONE)) is present, CKZ outputs
   /** commands to assist the manual copy process.
   /**
11 COPY TARGET-DB2(SSID(trss) LOC(tloc) -
      IP(ipaddr) -
      USR(uid) PASSWORD(password) -
      SYNCDB2-DDN(CKZSYNC) -
      SQLOUT-DDN(CKZSQL) -
      SIM(N) -
      DATA-MOVER( PGM(ADRDSSU) -
      FASTREP(PREF)) -
      REPLACE-TARGET-DSN(Y) -
      COPY-IF-NO-DB2-TARGET-OBJECTS(N) -
      OBJECT-TRANSLATE(CREATOR,ocs,oct)-
      AUTO-STOP-TARGET-SPACE(Y) -
      AUTO-START-TARGET-SPACE(Y) -
      FUZZY-COPY(N) -
      RESET-LOGRBA(Y)

```

Note: In this example, the following:

1. Execution of DB2 Cloning Tool Table Space Cloning main program.
2. DB2 Cloning Tool LOAD library must be authorized.
3. DD for the CKZINI member of PARMLIB. The CKZINI member contains the program variables.
4. CKZPRINT displays CKZINI tokens, control parameters, data set names and associated DB2 table spaces and index spaces, DB2 start and stop space command status and DFSMSdss program ADRDSSU commands and status.
5. CKZLOG - Displays LISTDEF processing and DB2 commands issued by DB2 Cloning Tool Table Space Cloning and responses/results of the commands. (This DD is not required.)
6. CKZLSTDF - DB2 Cloning Tool Table Space Cloning uses LISTDEF-like commands with standard IBM syntax to select the source table spaces and index spaces to be copied to the target. LISTDEF control statements can either

be entered into an existing PDS member (RECFM=FB,LRECL=80) and read as input using the CKZLSTDF DD statement, or can be read in-stream using CKZLSTDF DD *

7. DD for the CKZSYNC data set. CKZSYNC contains the parameters in an existing PDS member that will be used by the target job to make the VSAM object accessible on the target DB2 subsystem. The target job will be submitted by the user after the source job completes successfully.
8. DD for the CKZSQL data set. CKZSQL contains the SQL commands in an existing PDS member that will be used by the target job to synchronize identity column values between the target and source.
9. The SET LOCAL command specifies the local DB2 subsystem for the source job and the TCP/IP server job. DEFAULT-SQLID is applied to tables and indexes in LISTDEF without an explicit creator.
10. The optional HLQDDDF command may be used to pass input and/or output DDs to ADRDSSU. This may be useful to pass volsters to ADRDSSU for non-SMS output volume allocations.
11. The COPY command invokes the DFSMSdss program ADRDSSU to execute FlashCopy or SnapShot, or invokes the EMC API to execute TimeFinder/Clone, and prepares the object ID translation parameters for the target job. You may also specify the SIMULATE keyword with the COPY command.

Step 3: Set up and execute target job (CKZTRG)

The target job is comprised of a single step and a SYNCDB2 command for each data set copied to make the VSAM objects accessible on the target subsystem. It will optionally start the target table spaces and index spaces. The target job assumes the table spaces and index spaces are already stopped.

About this task

An example of the target JCL for this scenario can be found in the product JCL library member CKZTRG. The sample target job is valid for all four scenarios described. A detailed example is provided for illustration after the procedure steps. Each step contains a reference to the example.

Procedure

1. Specify the PARMLIB data set. Note that the DB2 Cloning Tool Table Space Cloning target job ignores fields it doesn't need. Those fields are TCPIP_OPTIONS and DSN_COPY_OPTIONS. This allows the same PARMLIB member to be used for the source, target, and TCP/IP server jobs. See line #3 in the target job example and the associated note 3.
2. Specify the input data set where the DB2 Cloning Tool Table Space Cloning source job created SQL to update the sequence number for identity columns. This can be found in the CKZSQL DD statement in the source job example, on line # 8. See line #6 in the target job example and the associated note 6.
3. Specify the input data set where the DB2 Cloning Tool Table Space Cloning source job created input parameters for the target job. This can be found in the CKZSYNC DD statement in the source job example, on line #7. See line #7 in the target job example and the associated note 7.

Target job example

For completeness and to illustrate where JCL parameters and DB2 Cloning Tool Table Space Cloning control statements must match, the following JCL includes sample DB2 Cloning Tool Table Space Cloning control statements.

The target job JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```
//jobcard
/*JOBPARM S=trgsys
/******
/*      COPYRIGHT 2009 MAINSTAR SOFTWARE CORPORATION      *
/*      ALL RIGHTS RESERVED                                *
/******
/*
/* Scenarios 1-4 - the target job is the same for all 4 scenarios.
/* The only change is the data set specified in CKZSQL and CKZIN.
/* These are output from the source job.
/*
/* The target job must be run on the target DB2 system.
/*
/* variables to be filled in ...
/*
/* jobcard - job card
/* trgsys - name of target system
/* hlq1 - CKZ LOADLIB and PARMLIB high level qualifier
/* hlq2 - SYSDUMP high level qualifier
/* dsnvr1 - DB2 high level qualifier
/* parmbr - CKZ PARMLIB member
/* dumpsc - SYSDUMP STORCLAS if needed
/* dumpmc - SYSDUMP MGMTCLAS if needed
/* hlq3 - CKZ users high level qualifier
/* lstdmbr - LISTDEF member name
/*
/*
/* Initiate DFSMSdss to EXECUTE FLASHCOPY or SNAPSHOT
/*
1 //STEP1 EXEC PGM=CKZ00500,REGION=0M
2 //STEPLIB DD DISP=SHR,DSN=hlq1.SCKZLOAD
// DD DISP=SHR,DSN=dsnvr1.SDSNEXIT
// DD DISP=SHR,DSN=dsnvr1.SDSNLOAD
/*
/*
/* CKZINI - Reads the CKZ parmlib to get default settings
/*
3 //CKZINI DD DISP=SHR,DSN=hlq1.SCKZPARAM(parmbr)
/*
/*
/* CKZPRINT - primary message output, required
/*
4 //CKZPRINT DD SYSOUT=*
/*
/*
/* CKZLOG - detailed message output, optional
/*
5 //CKZLOG DD SYSOUT=*
/*
/*
/* dump DDs
/*
/**SYSUDUMP DD SYSOUT=* Note: SYSDUMP preferred
/**SYSDUMP DD DSN=hlq2.SYSDUMP,DISP=(NEW,CATLG),
/* STORCLAS=dumpsc,MGMTCLAS=dumpmc,
/* SPACE=(CYL,(50,30),RLSE),
```

```

        /**          DCB=(LRECL=4160,BLKSIZE=4160,RECFM=FB),UNIT=SYSDA
        /**ABNLIGNR DD  DUMMY          do not remove if using ABENDAID
        /**
        /**
        /** CKZSQL - output ALTER TABLE SQL needed to synchronize
        /**
        /** identity column values between target and source.
6  /**CKZSQL  DD  DISP=SHR,DSN=h1q3.SQLOUT(1stdmbr)
        /**
        /**
        /** CKZIN - input commands that will be used by the target
        /** job to make the VSAM object(s) accessible on the target
        /** DB2 subsystem. Created by the source job.
        /**
7  /**CKZIN   DD  DISP=SHR,DSN=h1q3.SYNCDB2(1stdmbr)
        /**
        /*******
        /** sample control statements as would appear in SYNCDB2 member
        /** Note: all lines have /** added in column 1 for JCL compatibility.
        /*******
        /**
        /**CKZIN   DD  *
        /** 06313 09:16:18.55 JOBNAME=JMX8SRC JOBID=J0040889 */
        /**          SOURCE SUBSYSTEM=DB8G TARGET SUBSYSTEM=V81S */
        /**
        /***SET TRGJOB(Y) LSSID(trss) SQLDD(CKZSQL) SCANO(N)
        /**
        /**
        /*** TABLE SPACES - COPIED SUCCESSFULLY */
        /**
        /**
        /***SYNCDB2 TARGET-SSID (V81S) -
        /** TARGET-DSN (DSN081D.DSNDBC.ZTNDB.ZTN1TSS1.I0001.A001) -
        /** SPACE-TYPE (TS ) -
        /** PAGE-SIZE (4) -
        /** VCAT (DSN081D) -
        /** STOGROUP (ZSG884) -
        /** START-SPACE (Y) -
        /** RESET-LOGRBA (Y) -
        /** XLATE (DBID,X'012B,X'0113, -
        /**          PSID,X'0002,X'0047, -
        /**          TSOB,X'0001,X'0046, -
        /**          TBOB,X'0003,X'0048, -
        /**          TBOB,X'0008,X'004D, -
        /**          TBOB,X'000D,X'0052)
        /**
        /***SYNCDB2 TARGET-SSID (V81S) -
        /** TARGET-DSN (DSN081D.DSNDBC.ZTNDB.ZTN3TSN1.I0001.A001) -
        /** SPACE-TYPE (TS ) -
        /** PAGE-SIZE (4) -
        /** VCAT (DSN081D) -
        /** STOGROUP (ZSG884) -
        /** START-SPACE (Y) -
        /** RESET-LOGRBA (Y) -
        /** XLATE (DBID,X'0120,X'0113, -
        /**          PSID,X'0024,X'0013, -
        /**          TSOB,X'0023,X'0012, -
        /**          TBOB,X'0025,X'0014, -
        /**          TBOB,X'002A,X'0019, -
        /**          TBOB,X'002F,X'001E)
        /**
        /***SYNCDB2 TARGET-SSID (V81S) -
        /** TARGET-DSN (DSN081D.DSNDBC.ZTNDB.ZTN5TSN1.I0001.A001) -
        /** SPACE-TYPE (TP ) -
        /** PAGE-SIZE (4) -
        /** VCAT (DSN081D) -
        /** STOGROUP (ZSG884) -

```

```

/** START-SPACE (Y) -
/** RESET-LOGRBA (Y) -
/** XLATE (DBID,X'0120,X'0113, -
/**         PSID,X'0046,X'0035, -
/**         TSOB,X'0045,X'0034, -
/**         TBOB,X'0047,X'0036)
/**
/**SYNCDB2 TARGET-SSID (V81S) -
/** TARGET-DSN (DSN081D.DSNDBC.ZTNDB.ZTN5TSN1.I0001.A002) -
/** SPACE-TYPE (TP ) -
/** PAGE-SIZE (4) -
/** VCAT (DSN081D) -
/** STOGROUP (ZSG884) -
/** START-SPACE (Y) -
/** RESET-LOGRBA (Y) -
/** XLATE (DBID,X'0120,X'0114, -
/**         PSID,X'0046,X'0035, -
/**         TSOB,X'0045,X'0034, -
/**         TBOB,X'0047,X'0036)

```

Note: In this example, the following:

1. Execution of DB2 Cloning Tool Table Space Cloning main program.
2. DB2 Cloning Tool LOAD library must be authorized.
3. DD for the CKZINI member of PARMLIB. The CKZINI member contains the program variables.
4. CKZPRINT - Displays CKZINI tokens, CKZIN control parameters, DB2 SQL execution status, and SYNCDB2 status and START DB2 command status for each data set processed.
5. CKZLOG - Displays the DB2 commands issued by DB2 Cloning Tool Table Space Cloning and responses/results of the commands and detailed information about each DB2 page access. Not required.
6. DD for CKZSQL - The CKZSQL data set contains the SQL commands used by the target job to synchronize identity column values between the target and source.
7. DD for CKZIN - The CKZIN data set is where the source job created the input parameters for the target job. The source job DD statement is CKZSYNC DD. The sample job has typical target job input from CKZIN commented out.

Setup for scenario 2: FlashCopy, SnapShot, or TimeFinder/Clone - Some or all target table spaces and index spaces do not exist in DB2 catalog

Ideally, all target DB2 table spaces and index spaces will exist before the DB2 Cloning Tool Table Space Cloning source job is executed. However, if there is a narrow maintenance window for the copy and there is not enough time to fit creating the target table spaces and index spaces within the window, then use this procedure.

You may also use the procedure (shown in the following table) for the table spaces and index spaces that are inadvertently missed. The DB2 Cloning Tool Table Space Cloning source job will process existing and nonexistent target table spaces and index spaces in the same run. Only the data sets missing on the target require CKZFIX to be run.

Table 30. DB2 Cloning Tool Table Space Cloning - Setting up for using FlashCopy, SnapShot, or TimeFinder/Clone when some or all target objects do not exist

FlashCopy, SnapShot, or TimeFinder/Clone Setup Steps: when some or all target objects do not exist
“Step 1: Create target DB2 table spaces and index spaces in the target DB2 catalog (if required).”
<p>“Step 2: Set up and execute source job (CKZ2SRC)” on page 189. After you submit the source job and DB2 Cloning Tool Table Space Cloning detects missing target table spaces and index spaces, DB2 Cloning Tool Table Space Cloning will:</p> <ul style="list-style-type: none"> • Modify the target data set name(s) that it creates (the first character of the fifth node will be changed from “I” or “J” to “F”. DB2 will not recognize the new data set name.) • Generate IDCAMS parameters to delete the VSAM object(s) which will, later, be created by the target DB2. • Generate IDCAMS parameters to rename the “F” data sets to a name recognizable to DB2.
“Step 3 (Optional): Create missing target table spaces and index spaces” on page 193.
“Step 4: Set up and execute optional FIX job (CKZFIX)” on page 194.
“Step 5: Set up for a re-run of the source job” on page 195.
“Step 6: Set up and execute target job (CKZTRG)” on page 195.

Step 1: Create target DB2 table spaces and index spaces in the target DB2 catalog (if required)

The first step when copying by data set using FlashCopy, SnapShot, or TimeFinder/Clone when some or all target table spaces and index spaces do not exist is to create the target objects.

DB2 associates DB2 table spaces and index spaces with specific VSAM data set names. For example:

VCAT.DSNDBD.DBTEST1.TESTTBL.I0001.A001

where:

Node 1: VCAT - the data set high level qualifier

Node 2: DSNDBD is the data portion. DSNDBC is the cluster portion

Node 3: DBTEST1 - the database name in the DB2 catalog

Node 4: TESTTBL - the table space or index space name in the DB2 catalog

Node 5: Either I0001 or J0001 - a switch indicator

Node 6: A001 to E094A - a partition number or DB2 extent number

The correctly named DB2 table spaces and index spaces must exist in the target DB2 catalog for DB2 to recognize and use the VSAM data sets cloned to the target by DB2 Cloning Tool Table Space Cloning.

When DB2 creates VSAM objects, it puts internal object identifiers inside of them. Those internal object identifiers are probably not the same on the target as the internal object identifiers in the VSAM data sets copied from the source. In particular, the database ID will almost never be the same; however, the other IDs are relative to the database and will frequently be the same. DB2 Cloning Tool Table Space Cloning will have to translate the internal identifiers from the source

to those of the target to make the cloned VSAM data sets usable by the target DB2. Therefore you must create those target table spaces and index spaces if they do not exist.

Create as many of the following as are required by your application:

- Create the target stogroup(s)
- Create the target database(s)
- Create the target table space(s)
- Create the target table(s)
- Create the target index space(s)
- Create any target primary keys
- Create any target foreign keys
- Create any target alias(es), view(s), synonym(s), constraint(s), trigger(s), stored procedure(s), etc.

Note: DB2 Cloning Tool Table Space Cloning can be used to find target table spaces and index spaces that do not exist. Submit the source job with PGM(NONE) and copy parameter COPY-IF-NO-DB2-TARGET-OBJECTS(N). DB2 Cloning Tool Table Space Cloning will issue a warning message each time a source table space or index space cannot be paired with a target table space or index space.

Step 2: Set up and execute source job (CKZ2SRC)

The source job is comprised of several commands and phases to control the selection of the DB2 table spaces and index spaces on the source subsystem and to make them accessible on the target subsystem.

About this task

An example of the source JCL for this scenario can be found in the product JCL library member CKZ2SRC. A detailed example is provided for illustration after the procedure steps. Each step contains a reference to the example.

Procedure

1. Identify the source table spaces and index spaces to DB2 Cloning Tool Table Space Cloning using the DB2 Cloning Tool Table Space Cloning LISTDEF command.
 - The LISTDEF control statements can be either be entered into an existing PDS member (RECFM=FB,LRECL=80) and read as input by using the CKZLSTDF DD statement. See line #6 in the source job example and the associated note 6.
 - Or, the LISTDEF control statements can be read in-stream by using CKZLSTDF DD. See line #6 in the source job example and the associated note 6.
2. Specify the output data set where DB2 Cloning Tool Table Space Cloning will build the input parameters for the target job. See line #7 in the source job example and the associated note 7.
3. Identify the source DB2 subsystem in the SET command, keyword LOCAL-SSID(n), where n is the name of your source DB2 subsystem. See line #9 in the source job example and the associated note 9.
4. If input and/or output DDs must be passed to ADRDSSU, use the HLQDDDF command to specify those DDs. Identify the HLQ (high level qualifier) and a DD name or names using keyword HLQNAME(n) and DDNAME(dd1...,ddn),

where n is your high level qualifier and dd1...ddn is one or more DDs. See line #10 in the source job example and the associated note 10.

5. Identify the target DB2 subsystem information and copy method in the COPY command.
 - Identify the target DB2 subsystem to CKZIN in the COPY command, keyword TARGET-DB2(SSID(n) where n is the name of your target DB2 subsystem. See line #11 in the source job example and the associated note 11.
 - If using data set level FlashCopy or SnapShot to copy table spaces and index spaces, and your environment is set up to support data set level replication, use COPY command keyword, DATA-MOVER(PGM(ADRDSSU)) which invokes DFSMSdss program ADRDSSU to execute either FlashCopy or SnapShot to do the data set level copy operation. If FlashCopy is not available, ADRDSSU will substitute a normal copy operation, unless REQUIRED is specified. See line #11 in the source job example and the associated note 11. If using EMC TimeFinder/Clone, specify COPY command keyword DATA-MOVER(PGM(EMCAPI)).
 - Specify COPY parm COPY-IF-NO-DB2-TARGET-OBJECTS(Y) to prevent DB2 Cloning Tool Table Space Cloning from issuing a warning for each source table space and index space without a corresponding target table space or index space. See line #11 in the source job example and the associated note 11.
 - Specify COPY parm DEFVCAT to provide DB2 Cloning Tool Table Space Cloning with the space data set high level qualifier for all the table spaces and index spaces that do not exist in the target catalog. Note that only one can be supplied. If the target data sets have different high level qualifiers, run multiple DB2 Cloning Tool Table Space Cloning source jobs, each with the correct DEFVCAT value. See line #11 in the source job example and the associated note 11.

Results

CAUTION:

If COPY command keyword FUZZY-COPY(Y) is specified, then DB2 Cloning Tool Table Space Cloning will not stop the source table spaces and index spaces. However, we *do not recommend* this, because if the table spaces and index spaces are in RW status, there are data integrity issues.

Source job example

For completeness and to illustrate where JCL parameters and DB2 Cloning Tool Table Space Cloning control statements must match, the following JCL includes sample DB2 Cloning Tool Table Space Cloning control statements.

The source job JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```
//jobcard
/*JOBPARM S=srcsys
//*
//*****
//*      COPYRIGHT 2009 MAINSTAR SOFTWARE CORPORATION      *
//*      ALL RIGHTS RESERVED                                *
//*****
//*
//* Scenario 2 - one or more source spaces do not exist on the target
//*              and CKZ controls the copy process
```

```

/**
/** variables to be filled in ...
/**
/** jobcard - job card
/** srcsys - name of source system
/** hlq1 - CKZ LOADLIB and PARMLIB high level qualifier
/** hlq2 - SYSMDUMP high level qualifier
/** dsnvr1 - DB2 high level qualifier
/** parmbr - CKZ PARMLIB member
/** dumpsc - SYSMDUMP STORCLAS if needed
/** dumpmc - SYSMDUMP MGMTCLAS if needed
/** hlq3 - CKZ users high level qualifier
/** lstdmbr - LISTDEF member name
/** trss - target DB2 subsystem
/** dddd - DD DISP value if needed (must enable)
/** uuuu - DD UNIT value if needed (must enable)
/** vvvvvv - DD VOLUME value if needed (must enable)
/** srss - source DB2 subsystem
/** defsqlid - default SQLID applied to LISTDEF objects if needed
/** tloc - target subsystem if using DDF (delete if not needed)
/** ipaddr - target IP if using TCPIP (delete if not needed)
/** uid - user id if using DDF (delete if not needed)
/** password - password if using DDF (delete if not needed)
/** hlqtrgob - VCAT for target objects that do not exist
/** ocs - object creator on source DB2 (delete if not needed)
/** oct - object creator on target DB2 (delete if not needed)
/**
/**
/** Initiate DFSMSdss to EXECUTE FLASHCOPY or SNAPSHOT
/**
1 //STEP1 EXEC PGM=CKZ00500,REGION=0M
2 //STEPLIB DD DISP=SHR,DSN=hlq1.SCKZLOAD
// DD DISP=SHR,DSN=dsnvr1.SDSNEXIT
// DD DISP=SHR,DSN=dsnvr1.SDSNLOAD
/**
/**
/** CKZINI - Reads the CKZ parmlib to get default settings
/**
3 //CKZINI DD DISP=SHR,DSN=hlq1.SCKZPARAM(parmbr)
/**
/**
/** CKZPRINT - primary message output, required
/**
4 //CKZPRINT DD SYSOUT=*
/**
/**
/** CKZLOG - detailed message output, optional
/**
5 //CKZLOG DD SYSOUT=*
/**
/**
/** dump DDs
/**
/**SYSUDUMP DD SYSOUT=* Note: SYSMDUMP preferred
/**SYSMDUMP DD DSN=hlq2.SYSMDUMP,DISP=(NEW,CATLG),
/** STORCLAS=dumpsc,MGMTCLAS=dumpmc,
/** SPACE=(CYL,(50,30),RLSE),
/** DCB=(LRECL=4160,BLKSIZE=4160,RECFM=FB),UNIT=SYSDA
/**ABNLIGNR DD DUMMY do not remove if using ABENDAID
/**
/**
/** CKZLSTDF - CKZ uses LISTDEF like commands with standard IBM syntax
/** to select the source spaces to be copied to the target.
/**
6 //CKZLSTDF DD DISP=SHR,DSN=hlq3.LISTDEF(lstdmbr)
/**
/**

```

```

    /* CKZSYNC - output commands that will be used by the target
    /* job to make the VSAM object(s) accessible on the target
    /* DB2 subsystem.
    /*
7 //CKZSYNC DD DISP=OLD,DSN=h1q3.SYNCDDB2(1stdmbr)
    /*
    /*
    /* CKZSQL - output ALTER TABLE SQL needed to synchronize
    /* identity column values between target and source.
    /*
8 //CKZSQL DD DISP=OLD,DSN=h1q3.SQLOUT(1stdmbr)
    /*
    /*
    /* CKZIDCAM - output IDCAMS deletes and renames for all data sets
    /* copied to F0001, rather than I0001 or J0001.
    /*
    /*CKZIDCAM DD DISP=OLD,DSN=h1q3.IDCAMS(1stdmbr)
    /*
    /*
    /* CKZSTPT - output DB2 stop commands for all target spaces
    /*
    /*CKZSTPT DD DISP=OLD,DSN=h1q3.CMDSSTPT(1stdmbr)
    /*
    /*
    /* DSS output DD if required, see HLQDDDF
    /*
    /*MYOUTDD DD DISP=dddd,UNIT=uuuu,VOL=SER=(vvvvvv)
    /*
    /******
    /* sample control statements
    /******
    /*
    /*CKZIN DD *
    /*
    /*
    /* Source DB2 subsystem name and default SQLID if needed
    /*
9 SET LOCAL-SSID(srss) DEFAULT-SQLID(defsqlid)
    /*
    /*
    /* CKZ sends this DD name to ADRDSSU as output (enable if needed)
    /*
10 /* HLQDDDF HLQNAME(trgh1q) DDNAME(MYOUTDD)
    /*
    /*
    /* The COPY command invokes DFSMSdss program, ADRDSSU, to
    /* execute FlashCopy or Snapshot and prepares the object ID
    /* translation parameters for the target job. Or, if COPY
    /* keyword DATA-MOVER(PGM(NONE)) is present, CKZ outputs
    /* commands to assist the manual copy process.
    /*
    /*
11 COPY TARGET-DB2(SSID(trss) LOC(tloc) -
    IP(ipaddr) -
    USR(uid) PASSWORD(password) -
    DEFVCAT(h1qtrgob) -
    SYNCDDB2-DDN(CKZSYNC) -
    SQLOUT-DDN(CKZSQL) -
    IDCAMS-DDN(CKZIDCAM) -
    STOP-TARGET-DDN(CKZSTPT) -
    SIM(N) -
    DATA-MOVER( PGM(ADRDSSU) -
    FASTREP(PREF)) -
    REPLACE-TARGET-DSN(Y) -
    COPY-IF-NO-DB2-TARGET-OBJECTS(Y) -
    OBJECT-TRANSLATE(CREATOR,ocs,oct)-

```

AUTO-STOP-TARGET-SPACE(Y)	-
AUTO-START-TARGET-SPACE(Y)	-
FUZZY-COPY(N)	-
RESET-LOGRBA(Y)	

Note: In this example, the following:

1. Execution of DB2 Cloning Tool Table Space Cloning main program.
2. DB2 Cloning Tool LOAD library must be authorized.
3. DD for the CKZINI member of PARMLIB. The CKZINI member contains the program variables.
4. CKZPRINT displays CKZINI tokens, control parameters, data set names and associated DB2 table spaces and index spaces, DB2 start and stop space command status and DFSMSdss program ADRDSSU commands and status.
5. CKZLOG - Displays LISTDEF processing and DB2 commands issued by DB2 Cloning Tool Table Space Cloning and responses/results of the commands. (This DD is not required.)
6. CKZLSTDF - DB2 Cloning Tool Table Space Cloning uses LISTDEF-like commands with standard IBM syntax to select the source table spaces and index spaces to be copied to the target. LISTDEF control statements can either be entered into an existing PDS member (RECFM=FB,LRECL=80) and read as input using the CKZLSTDF DD statement, or can be read in-stream using CKZLSTDF DD *
7. DD for the CKZSYNC data set. CKZSYNC contains the parameters in an existing PDS member that will be used by the target job to make the VSAM object accessible on the target DB2 subsystem. The target job will be submitted by the user after the source job completes successfully.
8. DD for the CKZSQL data set. CKZSQL contains the SQL commands in an existing PDS member that will be used by the target job to synchronize identity column values between the target and source.
9. The SET LOCAL command specifies the local DB2 subsystem for the source job and the TCP/IP server job. DEFAULT-SQLID is applied to tables and indexes in LISTDEF without an explicit creator.
10. The optional HLQDDDF command may be used to pass input and/or output DDs to ADRDSSU. This may be useful to pass volsters to ADRDSSU for non-SMS output volume allocations.
11. The COPY command invokes the DFSMSdss program ADRDSSU to execute FlashCopy or SnapShot and prepares the object ID translation parameters for the target job.

Step 3 (Optional): Create missing target table spaces and index spaces

As the target table spaces and index spaces did not exist when the DB2 Cloning Tool Table Space Cloning source in step 2 was run, there are no STOP commands for the newly created table spaces and index spaces. If needed, at this point another DB2 Cloning Tool Table Space Cloning source job can be run to output these stop commands.

Use PGM(NONE) and no stops will be issued to the existing source spaces. Additionally, delete the IDCAMS-DDN parameter from the source job. If this delete is not performed, the correct IDCAMS-DDN data set will be overwritten. For all missing target table spaces and index spaces, the next step will delete these data sets and rename the copied data sets from F0001 to I0001.

Step 4: Set up and execute optional FIX job (CKZFIX)

The purpose of the FIX job is to facilitate making the data sets copied to the target subsystem available to the target DB2.

When DB2 Cloning Tool Table Space Cloning detects missing target DB2 table spaces and index spaces, parameters are written to data sets which will stop the missing table spaces and index spaces (after the user creates them), delete the new VSAM objects and rename the VSAM objects copied by DB2 Cloning Tool Table Space Cloning.

Sample JCL for the FIX job is in the product JCL library, member CKZFIX. Set up the job to point to the same data sets as STOP-TARGET-DDN and IDCAMS-DDN.

The CKZFIX job JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```
//jobcard
1 /*JOBPARM S=trgsys
  /*
  /******
  /*      COPYRIGHT 2009 MAINSTAR SOFTWARE CORPORATION      *
  /*      ALL RIGHTS RESERVED                                *
  /******
  /*
  /** Use when missing DB2 objects created after the copy.
  /**
  /** variables to be filled in ...
  /**
  /**   jobcard - job card
  /**   trgsys  - name of target system
  /**   dsnvr1 - DB2 high level qualifier
  /**   hlq3   - CKZ users high level qualifier
  /**   lstdmbr - LISTDEF member name
  /**
  /**
  /******
  /** Stop the DB2 created table space(s) AND index(es) *
  /** for the deletion and rename step to follow.      *
  /******
  /**
2 //STOP      EXEC PGM=IKJEFT01
  /*
  /**
  /*STEPLIB DD DISP=SHR,DSN=dsnvr1.SDSNLOAD
  /*SYSPRINT DD SYSOUT=*
  /*SYSTSPRT DD SYSOUT=*
  /**
  /**
  /** SYSTSIN DD was CKZSTPT DD in the source job CKZ2SRC
  /**
3 //SYSTSIN DD DISP=SHR,DSN=hlq3.CMDSSTPT(lstdmbr)
4 /** DSN      SYSTEM(trss)
  /** -STOP DATABASE(DBASEA) SPACE(TSPACEA)
  /** -STOP DATABASE(DBASEA) SPACE(INDEXA)
  /**
  /**
  /******
  /** Delete the DB2 created VSAM data set(s) and      *
  /** rename the cloned data set(s)                    *
  /******
  /**
5 //DELRNAME EXEC PGM=IDCAMS,COND=(4,LT)
  /*SYSPRINT DD SYSOUT=*
```

```

        /**
        /**
        /** SYSIN DD was CKZIDCAM DD in the source job CKZ2SRC
        /**
6 //SYSIN DD DISP=SHR,DSN=h1q3.IDCAMS(1stdmbr)
        /**
        /** DELETE DSN081D.DSNDBC.DBASEA.TSPACEA.I0001.A001 -
        /** CLUSTER NOERASE PURGE
        /** ALTER -
        /** DSN081D.DSNDBC.DBASEA.TSPACEA.F0001.A001 -
        /** NEWNAME(DSN081D.DSNDBC.DBASEA.TSPACEA.I0001.A001)
        /** ALTER -
        /** DSN081D.DSNDBD.DBASEA.TSPACEA.F0001.A001 -
        /** NEWNAME(DSN081D.DSNDBD.DBASEA.TSPACEA.I0001.A001)
        /**

```

Note: In this example, the following:

1. JOBPARM where *trgsys* is the target z/OS system.
2. Execution of TSO monitor program.
3. SYSTSIN – Data set referenced by source job COPY command keyword STOP-TARGET-DDN. This data set is only available if the source job was resubmitted after the missing target table spaces and index spaces have been created.
4. SYSTSIN – DSN command. *trss* is the target SSID. This is supplied by DB2 Cloning Tool Table Space Cloning in the output data set.
5. Execution of IDCAMS – To delete the VSAM objects created by the target DB2 when the table spaces and index spaces were created and rename the VSAM objects copied from the source DB2.
6. SYSIN – Data set referenced by source job COPY command keyword IDCAMS-DDN.

Step 5: Set up for a re-run of the source job

Re-running the source job at this point will capture the object IDs of the newly created DB2 table spaces and index spaces and update the CKZSYNC and CKZSQL (if required) data sets used as input for the SYNCDB2 command.

The COPY command keyword must be DATA-MOVER(PGM(NONE)). With PGM(NONE), no DB2 stop or start space commands will be issued.

Step 6: Set up and execute target job (CKZTRG)

The target job is comprised of a single step and SYNCDB2 command to make the VSAM objects accessible on the target subsystem. The target job assumes all target spaces are stopped. It will optionally start the target table spaces and index spaces.

About this task

An example of the target JCL for this scenario can be found in the product JCL library member CKZTRG. A detailed example is provided for illustration after the procedure steps. Each step contains a reference to the example and associated note.

Procedure

1. Specify the PARMLIB data set. Note that the DB2 Cloning Tool Table Space Cloning target job ignores fields it doesn't need. Those fields are TCPIP_OPTIONS and DSN_COPY_OPTIONS. This allows the same PARMLIB member to be used for the source, target, and TCP/IP server jobs. See line #3 in the target job example and the associated note 3.

2. Specify the input data set where the DB2 Cloning Tool Table Space Cloning source job created SQL to update the sequence number for identity columns. This can be found in the CKZSQL DD statement in the source job example, on line #8. See line #6 in the target job example and the associated note 6.
3. Specify the input data set where the DB2 Cloning Tool Table Space Cloning source job created input parameters for the target job. This can be found in the CKZSYNC DD statement in the source job example, on line #7. See line #7 in the target job example and the associated note 7.

Results

Target job example

For completeness and to illustrate where JCL parameters and DB2 Cloning Tool Table Space Cloning control statements must match, the following JCL includes sample DB2 Cloning Tool Table Space Cloning control statements.

The target job JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

//jobcard
/*JOBPARM S=trgsys
//*****
/*      COPYRIGHT 2009 MAINSTAR SOFTWARE CORPORATION      *
/*      ALL RIGHTS RESERVED                                *
//*****
//*
//* Scenarios 1-4 - the target job is the same for all 4 scenarios.
//* The only change is the data set specified in CKZSQL and CKZIN.
//* These are output from the source job.
//*
//* The target job must be run on the target DB2 system.
//*
//* variables to be filled in ...
//*
//* jobcard - job card
//* trgsys - name of target system
//* hlq1 - CKZ LOADLIB and PARMLIB high level qualifier
//* hlq2 - SYSMDUMP high level qualifier
//* dsnvr1 - DB2 high level qualifier
//* parmbr - CKZ PARMLIB member
//* dumpsc - SYSMDUMP STORCLAS if needed
//* dumpmc - SYSMDUMP MGMTCLAS if needed
//* hlq3 - CKZ users high level qualifier
//* lstdmbr - LISTDEF member name
//*
//*
//* Initiate DFSMSdss to EXECUTE FLASHCOPY or SNAPSHOT
//*
1 //STEP1 EXEC PGM=CKZ00500,REGION=0M
2 //STEPLIB DD DISP=SHR,DSN=hlq1.SCKZLOAD
// DD DISP=SHR,DSN=dsnvr1.SDSNEXIT
// DD DISP=SHR,DSN=dsnvr1.SDSNLOAD
//*
//*
//* CKZINI - Reads the CKZ parmlib to get default settings
//*
3 //CKZINI DD DISP=SHR,DSN=hlq1.SCKZPARAM(parmbr)
//*
//*
//* CKZPRINT - primary message output, required
//*
4 //CKZPRINT DD SYSOUT=*

```



```

/**
/**
/** CKZLOG - detailed message output, optional
/**
5 //CKZLOG DD SYSOUT=*
/**
/**
/** dump DDs
/**
/**SYSUDUMP DD SYSOUT=* Note: SYSDUMP preferred
/**SYSDUMP DD DSN=h1q2.SYSDUMP,DISP=(NEW,CATLG),
/** STORCLAS=dumpscl,MGMTCLAS=dumpscl,
/** SPACE=(CYL,(50,30),RLSE),
/** DCB=(LRECL=4160,BLKSIZE=4160,RECFM=FB),UNIT=SYSDA
/**ABNLIGNR DD DUMMY do not remove if using ABENDAID
/**
/**
/** CKZSQL - output ALTER TABLE SQL needed to synchronize
/**
/** identity column values between target and source.
6 //CKZSQL DD DISP=SHR,DSN=h1q3.SQLOUT(1stdmbr)
/**
/**
/** CKZIN - input commands that will be used by the target
/** job to make the VSAM object(s) accessible on the target
/** DB2 subsystem. Created by the source job.
/**
7 //CKZIN DD DISP=SHR,DSN=h1q3.SYNCDB2(1stdmbr)
/**
/*******
/** sample control statements as would appear in SYNCDB2 member
/** Note: all lines have /** added in column 1 for JCL compatibility.
/*******
/**
/**CKZIN DD *
/** 06313 09:16:18.55 JOBNAME=JMX8SRC JOBID=J0040889 */
/** SOURCE SUBSYSTEM=DB8G TARGET SUBSYSTEM=V81S */
/**
/**SET TRGJOB(Y) LSSID(trss) SQLDD(CKZSQL) SCANO(N)
/**
/**
/** TABLE SPACES - COPIED SUCCESSFULLY */
/**
/**
/**SYNCDB2 TARGET-SSID (V81S) -
/** TARGET-DSN (DSN081D.DSNDBC.ZTNDB.ZTN1TSS1.I0001.A001) -
/** SPACE-TYPE (TS ) -
/** PAGE-SIZE (4) -
/** VCAT (DSN081D) -
/** STOGROUP (ZSG884) -
/** START-SPACE (Y) -
/** RESET-LOGRBA (Y) -
/** XLATE (DBID,X'0120,X'0113, -
/** PSID,X'0002,X'0047, -
/** TSOB,X'0001,X'0046, -
/** TBOB,X'0003,X'0048, -
/** TBOB,X'0008,X'004D, -
/** TBOB,X'000D,X'0052)
/**
/**SYNCDB2 TARGET-SSID (V81S) -
/** TARGET-DSN (DSN081D.DSNDBC.ZTNDB.ZTN3TSN1.I0001.A001) -
/** SPACE-TYPE (TS ) -
/** PAGE-SIZE (4) -
/** VCAT (DSN081D) -
/** STOGROUP (ZSG884) -
/** START-SPACE (Y) -
/** RESET-LOGRBA (Y) -

```

```

/** XLATE (DBID,X'0120,X'0113, -
/**      PSID,X'0024,X'0013, -
/**      TSOB,X'0023,X'0012, -
/**      TBOB,X'0025,X'0014, -
/**      TBOB,X'002A,X'0019, -
/**      TBOB,X'002F,X'001E)
/**
/**SYNCDB2 TARGET-SSID (V81S) -
/** TARGET-DSN (DSN081D.DSNDBC.ZTNDB.ZTN5TSN1.I0001.A001) -
/** SPACE-TYPE (TP ) -
/** PAGE-SIZE (4) -
/** VCAT (DSN081D) -
/** STOGROUP (ZSG884) -
/** START-SPACE (Y) -
/** RESET-LOGRBA (Y) -
/** XLATE (DBID,X'0120,X'0113, -
/**      PSID,X'0046,X'0035, -
/**      TSOB,X'0045,X'0034, -
/**      TBOB,X'0047,X'0036)
/**
/**SYNCDB2 TARGET-SSID (V81S) -
/** TARGET-DSN (DSN081D.DSNDBC.ZTNDB.ZTN5TSN1.I0001.A002) -
/** SPACE-TYPE (TP ) -
/** PAGE-SIZE (4) -
/** VCAT (DSN081D) -
/** STOGROUP (ZSG884) -
/** START-SPACE (Y) -
/** RESET-LOGRBA (Y) -
/** XLATE (DBID,X'0120,X'0113, -
/**      PSID,X'0046,X'0035, -
/**      TSOB,X'0045,X'0034, -
/**      TBOB,X'0041,X'0036)

```

Note: In this example, the following:

1. Execution of DB2 Cloning Tool Table Space Cloning main program.
2. DB2 Cloning Tool LOAD library must be authorized.
3. DD for the CKZINI member of PARMLIB. The CKZINI member contains the program variables.
4. CKZPRINT displays CKZINI tokens, CKZIN control parameters, DB2 SQL execution status, and SYNCDB2 status and START DB2 command status for each data set processed.
5. CKZLOG displays the DB2 commands issued by DB2 Cloning Tool Table Space Cloning and responses/results of the commands and detailed information about each DB2 page access. (not required)
6. DD for CKZSQL - The CKZSQL data set contains the SQL commands used by the target job to synchronize identity column values between the target and source.
7. DD for CKZIN - The CKZIN data set is where the source job created the input parameters for the target job. The source job DD statement is CKZSYNC DD. The sample job has typical target job input from CKZIN commented out.

Chapter 15. Set up procedures for copy by data set for all other methodologies

This topic contains two DB2 Cloning Tool Table Space Cloning setup procedures for you to use if data set copies are to be created using copy methodologies other than FlashCopy or SnapShot via DFSMSdss or TimeFinder/Clone. Choose which procedure to use based on the following scenarios.

Setup procedures for scenarios using FlashCopy, SnapShot, or TimeFinder/Clone are documented in Chapter 14, "Setup procedures for copy by data set with FlashCopy, SnapShot, or TimeFinder/Clone," on page 179.

Setup for scenario 3: Other copy methodologies - all target table spaces and index spaces already exist in DB2 catalog

Use the procedure in the following table for all other copy methodologies, other than FlashCopy, SnapShot, or TimeFinder/Clone, to copy source table spaces and index spaces to the target DB2 subsystem when the target table spaces and index spaces already exist in the DB2 catalog.

Table 31. DB2 Cloning Tool Table Space Cloning - Setting up for other copy methodologies

Setup steps for other methodologies
"Step 1: Verify/create target DB2 table spaces and index spaces in target DB2 catalog."
"Step 2: Set up and execute source job (CKZ3SRC)" on page 200.
"Step 3 (Optional): Set up and execute stop target job (CKZSTPT)" on page 204.
"Step 4 (Optional): Set up and execute stop source job (CKZSTPS)" on page 205.
"Step 5: Set up and execute non-DB2 Cloning Tool Table Space Cloning to execute data set copies outside DB2 Cloning Tool Table Space Cloning" on page 206.
"Step 6 (Optional): Set up and execute job to re-start source DB2 table spaces and index spaces (CKZSTRS)" on page 206.
"Step 7: Set up and execute target job (CKZTRG)" on page 207.

Step 1: Verify/create target DB2 table spaces and index spaces in target DB2 catalog

The first step when copying by data set using other methodologies to verify/create the target objects in the target DB2 catalog.

DB2 associates DB2 table spaces and index spaces with specific VSAM data set names. For example:

VCAT.DSNDBD.DBTEST1.TESTTBL.I0001.A001

where:

Node 1: VCAT - the data set high level qualifier

Node 2: DSNDBD is the data portion. DSNDBC is the cluster portion

Node 3: DBTEST1 - the database name in the DB2 catalog

Node 4: TESTTBL - the table space or index space name in the DB2 catalog

Node 5: Either I0001 or J0001 (a switch indicator)

Node 6: A001 to E094, a partition number or DB2 extent number

The correctly named DB2 table spaces and index spaces must exist in the target DB2 catalog for DB2 to recognize and use the VSAM data sets cloned to the target by DB2 Cloning Tool Table Space Cloning.

When DB2 creates VSAM objects, it puts internal object identifiers inside of them. Those internal object identifiers are probably not the same on the target as the internal object identifiers in the VSAM data sets cloned from the source. In particular, the database ID will almost never be the same; however, the other IDs are relative to the database and will frequently be the same. DB2 Cloning Tool Table Space Cloning will have to translate the internal identifiers from the source to those of the target to make the cloned VSAM data sets usable by the target DB2.

Create as many of the following as are required by your application:

- Create the target stogroup(s)
- Create the target database(s)
- Create the target table space(s)
- Create the target table(s)
- Create the target index space(s)
- Create any target primary keys
- Create any target foreign keys
- Create any target alias(es), view(s), synonym(s), constraint(s), trigger(s), stored procedure(s), etc.

Note: DB2 Cloning Tool Table Space Cloning can be used to find target table spaces and index spaces that do not exist. Submit the source job with PGM(NONE) and copy parameter COPY-IF-NO-DB2-TARGET-OBJECTS(N). DB2 Cloning Tool Table Space Cloning will issue a warning message each time a source table space or index space cannot be paired with a target table space or index space.

Step 2: Set up and execute source job (CKZ3SRC)

The source job is comprised of several commands and phases to control the selection of the DB2 table spaces and index spaces on the source subsystem and to make them accessible on the target subsystem.

About this task

An example of the source JCL for this scenario can be found in the product JCL library member CKZ3SRC. A detailed example is provided for illustration after the procedure steps. Each step contains a reference to the example.

Procedure

1. Identify the source table spaces and index spaces to DB2 Cloning Tool Table Space Cloning using the DB2 Cloning Tool Table Space Cloning LISTDEF command.
 - The LISTDEF control statements can be either be entered into an existing PDS member (RECFM=FB,LRECL=80) and read as input by using the CKZLSTDF DD statement. See line #6 in the source job example and the associated note 6.
 - Or, the LISTDEF control statements can be read in-stream by using CKZLSTDF DD. See line #6 in the source job example and the associated note 6.

2. Specify the output data set where DB2 Cloning Tool Table Space Cloning will build the input parameters for the target job. See line #7 in the source job example and the associated note 7.
3. Identify the source DB2 subsystem in the SET command, keyword LOCAL-SSID(*n*), where *n* is the name of your source DB2 subsystem. See line #9 in the source job example and the associated note 9.
4. Identify the target DB2 subsystem information and copy method in the COPY command.
 - Identify the target DB2 subsystem to CKZIN in the COPY command, keyword TARGET-DB2(SSID(*n*)) where *n* is the name of your target DB2 subsystem. See line #11 in the source job example and the associated note 11.
 - Specify keyword DATA-MOVER(PGM(NONE)) in the COPY command to indicate that the copy is to be done outside of DB2 Cloning Tool Table Space Cloning. See line #11 in the source job example and the associated note 11.

Results

Source job example

For completeness and to illustrate where JCL parameters and DB2 Cloning Tool Table Space Cloning control statements must match, the following JCL includes sample DB2 Cloning Tool Table Space Cloning control statements.

The source job JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

//jobcard
/*JOBPARM S=srcsys
/*
/******
/*      COPYRIGHT 2009 MAINSTAR SOFTWARE CORPORATION      *
/*      ALL RIGHTS RESERVED                                *
/******
/*
/* Scenario 3 - all spaces to be copied exist on the target and
/*              copies are done outside of CKZ
/*
/* variables to be filled in ...
/*
/* jobcard - job card
/* srcsys - name of source system
/* hlq1 - CKZ LOADLIB and PARMLIB high level qualifier
/* hlq2 - SYSDUMP high level qualifier
/* dsnvr1 - DB2 high level qualifier
/* parmbr - CKZ PARMLIB member
/* dumpsc - SYSDUMP STORCLAS if needed
/* dumpmc - SYSDUMP MGMTCLAS if needed
/* hlq3 - CKZ users high level qualifier
/* lstdmbr - LISTDEF member name
/* trss - target DB2 subsystem
/* dddd - DD DISP value if needed (must enable)
/* uuuu - DD UNIT value if needed (must enable)
/* vvvvv - DD VOLUME value if needed (must enable)
/* srss - source DB2 subsystem
/* defsqlid - default SQLID applied to LISTDEF objects if needed
/* tloc - target subsystem if using DDF (delete if not needed)
/* ipaddr - target IP if using TCPIP (delete if not needed)
/* uid - user id if using DDF (delete if not needed)
/* password - password if using DDF (delete if not needed)
/* ocs - object creator on source DB2 (delete if not needed)

```

```

    /**      oct      - object creator on target DB2 (delete if not needed)
    /**
    /**
    /** Initiate DFSMSdss to EXECUTE FLASHCOPY or SNAPSHOT
    /**
1 //STEP1      EXEC PGM=CKZ00500,REGION=0M
2 //STEPLIB   DD DISP=SHR,DSN=h1q1.SCKZLOAD
    //          DD DISP=SHR,DSN=dsnvr1.SDSNEXIT
    //          DD DISP=SHR,DSN=dsnvr1.SDSNLOAD
    /**
    /**
    /** CKZINI - Reads the CKZ parmlib to get default settings
    /**
3 //CKZINI    DD DISP=SHR,DSN=h1q1.SCKZPARAM(parmmbr)
    /**
    /**
    /** CKZPRINT - primary message output, required
    /**
4 //CKZPRINT DD SYSOUT=*
    /**
    /**
    /** CKZLOG - detailed message output, optional
    /**
5 //CKZLOG    DD SYSOUT=*
    /**
    /**
    /** dump DDs
    /**
    /**SYSDUMP DD SYSOUT=*          Note: SYSDUMP preferred
    /**SYSDUMP DD DSN=h1q2.SYSDUMP,DISP=(NEW,CATLG),
    /**          STORCLAS=dumpscl,MGMTCLAS=dumpmc,
    /**          SPACE=(CYL,(50,30),RLSE),
    /**          DCB=(LRECL=4160,BLKSIZE=4160,RECFM=FB),UNIT=SYSDA
    /**ABNLIGNR DD DUMMY          do not remove if using ABENDAID
    /**
    /**
    /** CKZLSTDF - CKZ uses LISTDEF like commands with standard IBM syntax
    /** to select the source spaces to be copied to the target.
    /**
6 //CKZLSTDF DD DISP=SHR,DSN=h1q3.LISTDEF(1stdmbr)
    /**
    /**
    /** CKZSYNC - output commands that will be used by the target
    /** job to make the VSAM object(s) accessible on the target
    /** DB2 subsystem.
    /**
7 //CKZSYNC   DD DISP=OLD,DSN=h1q3.SYNCDB2(1stdmbr)
    /**
    /**
    /** CKZCOPY - output list to copy with from data set name and to
    /** data set name
    /**
8 //CKZCOPY   DD DISP=OLD,DSN=h1q3.COPYDSNS(1stdmbr)
    /**
    /**
    /** CKZSQL - output ALTER TABLE SQL needed to synchronize
    /** identity column values between target and source.
    /**
9 //CKZSQL    DD DISP=OLD,DSN=h1q3.SQLOUT(1stdmbr)
    /**
    /**
    /** CKZSTPT - output DB2 stop commands for all target spaces
    /**
    /**CKZSTPT DD DISP=OLD,DSN=h1q3.CMDSSTPT(1stdmbr)
    /**
    /**
    /** CKZSTPS - output DB2 stop commands for all source spaces

```

```

/*
// CKZSTPS DD DISP=OLD,DSN=h1q3.CMDSSTPS(1stdmbr)
/*
/*
/* CKZSTRS - output DB2 stop commands for all source spaces
/*
//CKZSTRS DD DISP=OLD,DSN=h1q3.CMDSSTRS(1stdmbr)
/*
/*
//*****
/* sample control statements
//*****
/*
//CKZIN DD *
/* */
/* */
/* Source DB2 subsystem name and default SQLID if needed */
/* */
10 SET LOCAL-SSID(srss) DEFAULT-SQLID(defsqlid)
/* */
/* */
/* The COPY command prepares the objects ID translation */
/* parameters for the target job and outputs commands to */
/* assist the manual copy process. */
/* */
11 COPY TARGET-DB2(SSID(trss) LOC(tloc) -
      IP(ipaddr) -
      USR(uid) PASSWORD(password)) -
      DATASETS-TO-COPY-DDN(CKZCOPY) -
      SYNCDB2-DDN(CKZSYNC) -
      SLOUT-DDN(CKZSQL) -
      STOP-TARGET-DDN(CKZSTPT) -
      STOP-SOURCE-DDN(CKZSTPS) -
      START-SOURCE-DDN(CKZSTRS) -
      SIM(N) -
      DATA-MOVER( PGM(NONE)) -
      REPLACE-TARGET-DSN(Y) -
      COPY-IF-NO-DB2-TARGET-OBJECTS(N) -
      OBJECT-TRANSLATE(CREATOR,ocs,oct)-
      AUTO-STOP-TARGET-SPACE(Y) -
      AUTO-START-TARGET-SPACE(Y) -
      RESET-LOGRBA(Y)

```

Note: In this example, the following:

1. Execution of DB2 Cloning Tool Table Space Cloning main program.
2. DB2 Cloning Tool LOAD library must be authorized.
3. DD for the CKZINI member of PARMLIB. The CKZINI member contains the program variables.
4. CKZPRINT displays CKZINI tokens, control parameters, data set names and associated DB2 table spaces and index spaces, DB2 start and stop space command status and DFSMSdss program ADRDSSU commands and status.
5. CKZLOG - Displays LISTDEF processing and DB2 commands issued by DB2 Cloning Tool Table Space Cloning and responses/results of the commands. (This DD is not required.)
6. CKZLSTDF - DB2 Cloning Tool Table Space Cloning uses LISTDEF-like commands with standard IBM syntax to select the source table spaces and index spaces to be copied to the target. LISTDEF control statements can either be entered into an existing PDS member (RECFM=FB,LRECL=80) and read as input using the CKZLSTDF DD statement, or can be read in-stream using CKZLSTDF DD *

7. DD for the CKZSYNC data set. CKZSYNC contains the parameters in an existing PDS member that will be used by the target job to make the VSAM object accessible on the target DB2 subsystem. The target job will be submitted by the user after the source job completes successfully.
8. DD for CKZCOPY - The CKZCOPY data sets contains a list of input and output data sets if the keyword for the COPY command is: DATA-MOVER(PGM(NONE)). The purpose of this listing is to assist copying the selected VSAM objects outside of DB2 Cloning Tool Table Space Cloning.
9. DD for CKZSQL - The CKZCKZSQL data set contains the SQL commands in an existing PDS member that will be used by the target job to synchronize identity column values between the target and source.
10. The SET LOCAL command specifies the local DB2 subsystem for the source job and the TCP/IP server job. DEFAULT-SQLID is applied to tables and indexes in LISTDEF without an explicit creator.
11. COPY Command - The COPY command with DATA-MOVER(PGM(NONE)) assumes the copy of the data sets is to be done outside of DB2 Cloning Tool Table Space Cloning, and only prepares the object ID transaction parameters for the target job and outputs DB2 start and stop space commands and IDCAMS delete/define commands into data sets if requested.

Step 3 (Optional): Set up and execute stop target job (CKZSTPT)

The purpose of the CKZSTPT job is to facilitate copying data sets outside of DB2 Cloning Tool Table Space Cloning.

When the COPY command keyword is DATA-MOVER(PGM(NONE)), DB2 Cloning Tool Table Space Cloning writes DB2 stop parameters for the target DB2 table spaces and index spaces to a data set pointed to by COPY keyword STOP-TARGET-DDN(?).

Sample JCL for the CKZSTPT job is in the product JCL library, member CKZSTPT. Set up the job to point to the same data set as STOP-TARGET-DDN.

The CKZSTPT job JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

1 //jobcard
  /*JOBPARM S=trgsys
  /******
  /*      COPYRIGHT 2009 MAINSTAR SOFTWARE CORPORATION      *
  /*      ALL RIGHTS RESERVED                                *
  /******
  /*
  /** Use to stop target objects if a copy method outside CKZ
  /** is used to copy data sets from the source to the target.
  /**
  /** variables to be filled in ...
  /**
  /** jobcard - job card
  /** trgsys - name of target system
  /** dsnvr1 - DB2 high level qualifier
  /** hlq3 - CKZ users high level qualifier
  /** lstdmbr - LISTDEF member name
  /**
  /**
  /******
  /** Stop the DB2 target table and index spaces      *
```



```

//*****
//*
2 //STOPT EXEC PGM=IKJEFT01
//STEPLIB DD DISP=SHR,DSN=dsnvr1.SDSNLOAD
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//*
//*
//* SYSTSIN DD was CKZSTPT DD in the source job.
//*
3 //SYSTSIN DD DISP=SHR,DSN=h1q3.CMDSSTPT(1stdmbr)
4 //* DSN SYSTEM(trss)
//* -STOP DATABASE(DBASEA) SPACE(TSPACEA)
//* -STOP DATABASE(DBASEA) SPACE(INDEXA)
//*

```

Note: In this example, the following:

1. JOBPARM where *trgsys* is the target z/OS system.
2. Execution of TSO monitor program.
3. SYSTSIN – Data set referenced by source job COPY command keyword STOP-TARGET-DDN
4. SYSTSIN – DSN command. *trss* is the target SSID. This is supplied by DB2 Cloning Tool Table Space Cloning in the output data set.

Step 4 (Optional): Set up and execute stop source job (CKZSTPS)

The purpose of the CKZSTPS job is to facilitate copying data sets outside of DB2 Cloning Tool Table Space Cloning.

When the COPY command keyword is DATA-MOVER(PGM(NONE)), DB2 Cloning Tool Table Space Cloning writes DB2 stop parameters for the source DB2 table spaces and index spaces to a data set pointed to by COPY keyword STOP-SOURCE-DDN(ddname).

Sample JCL for the CKZSTPTS job is in the product JCL library, member CKZSTPTS. Set up the job to point to the same data set as STOP-SOURCE-DDN.

The CKZSTPTS job JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

//jobcard
1 /*JOBPARM S=srcsys
//*****
//* COPYRIGHT 2009 MAINSTAR SOFTWARE CORPORATION *
//* ALL RIGHTS RESERVED *
//*****
//*
//* Use to stop source spaces if a copy method outside CKZ
//* is used to copy data sets from the source to the target.
//*
//* variables to be filled in ...
//*
//* jobcard - job card
//* srcsys - name of source system
//* dsnvr1 - DB2 high level qualifier
//* h1q3 - CKZ users high level qualifier
//* 1stdmbr - LISTDEF member name
//*
//*
2 //STOPS EXEC PGM=IKJEFT01

```

```

//STEPLIB DD DISP=SHR,DSN=dsnvr1.SDSNLOAD
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//*
//*
//* SYSTSIN DD was CKZSTPS DD in the source job.
//*
3 //SYSTSIN DD DISP=SHR,DSN=h1q3.CMDSSTPS(1stdmbr)
4 //* DSN SYSTEM(srss)
//* -STOP DATABASE(DBASEA) SPACE(TSPACEA)
//* -STOP DATABASE(DBASEA) SPACE(INDEXA)
//*

```

Note: In this example, the following:

1. JOBPARM where *srcsys* is the source z/OS system.
2. Execution of TSO monitor program.
3. SYSTSIN – Data set referenced by source job COPY command keyword STOP-SOURCE-DDN
4. SYSTSIN – DSN command. *srss* is the source SSID. This is supplied by DB2 Cloning Tool Table Space Cloning in the output data set.

Step 5: Set up and execute non-DB2 Cloning Tool Table Space Cloning job to execute data set copies outside DB2 Cloning Tool Table Space Cloning

Employ whatever procedures and syntax appropriate to your chosen replication tool.

If requested by the COPY parameter DATASETS-TO-COPY-DDN, DB2 Cloning Tool Table Space Cloning writes out a list of the source and target VSAM object pairs. The target data set names will have the correct high level qualifier and the correct switch (I/J) indicator. In addition, only the qualifying data set pairs will be listed.

A sample output:

```

* 06330 11:56:36.79 JOBNAME=JMX8SRC JOBID=J0044593
* SOURCE SUBSYSTEM=DB8G TARGET SUBSYSTEM=V81S
*
*
* TABLE SPACE DATA SETS TO BE COPIED
*
SOURCE DATASET=DSN081C.DSNDBC.ZTNDB.ZTN1TSS1.I0001.A001
TARGET DATASET=DSN081D.DSNDBC.ZTNDB.ZTN1TSS1.I0001.A001
*
SOURCE DATASET=DSN081C.DSNDBC.ZTNDB.ZTN2TSSC.I0001.A001
TARGET DATASET=DSN081D.DSNDBC.ZTNDB.ZTN2TSSC.J0001.A001
*
SOURCE DATASET=DSN081C.DSNDBC.ZTNDB.ZTN3TSN1.I0001.A001
TARGET DATASET=DSN081D.DSNDBC.ZTNDB.ZTN3TSN1.J0001.A001

```

Step 6 (Optional): Set up and execute job to re-start source DB2 table spaces and index spaces (CKZSTRS)

The purpose of the CKZSTRS job is to facilitate copying data sets outside of DB2 Cloning Tool Table Space Cloning.

When the COPY command keyword is DATA-MOVER(PGM(NONE)), DB2 Cloning Tool Table Space Cloning writes DB2 start parameters for the source DB2 table spaces and index spaces to a data set pointed to by COPY keyword START-SOURCE-DDN(?).

Sample JCL for the CKZSTRS job is in the product JCL library, member CKZSTRS. Set up the job to point to the same data set as START-SOURCE-DDN.

The CKZSTRS job JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

//jobcard
1 /*JOBPARM S=srcsys
  /******
  /*      COPYRIGHT 2009 MAINSTAR SOFTWARE CORPORATION      *
  /*      ALL RIGHTS RESERVED                                *
  /******
  /*
  /** Use to start source objects if a copy method outside CKZ
  /** is used to copy data sets from the source to the target.
  /**
  /** variables to be filled in ...
  /**
  /** jobcard - job card
  /** srcsys - name of source system
  /** dsnvr1 - DB2 high level qualifier
  /** hlq3 - CKZ users high level qualifier
  /** lstdmbr - LISTDEF member name
  /**
  /**
2 //STARTS EXEC PGM=IKJEFT01
  //STEPLIB DD DISP=SHR,DSN=dsnvr1.SDSNLOAD
  //SYSPRINT DD SYSOUT=*
  //SYSTSPRT DD SYSOUT=*
  /**
  /**
  /** SYSTSIN DD was CKZSTRS DD in the source job.
  /**
3 //SYSTSIN DD DISP=SHR,DSN=hlq3.CMDSSTRS(lstdmbr)
4 /** DSN SYSTEM(srss)
  /** -START DATABASE(DBASEA) SPACE(TSPACEA)
  /** -START DATABASE(DBASEA) SPACE(INDEXA)
  /**

```

Note: In this example, the following:

1. JOBPARM where *srcsys* is the source z/OS system.
2. Execution of TSO monitor program.
3. SYSTSIN – Data set referenced by source job COPY command keyword START-SOURCE-DDN
4. SYSTSIN – DSN command. *srss* is the source SSID. This is supplied by DB2 Cloning Tool Table Space Cloning in the output data set.

Step 7: Set up and execute target job (CKZTRG)

The target job is comprised of a single step and SYNCDB2 command to make the VSAM objects accessible on the target subsystem. It will optionally stop and start the target table spaces and index spaces.

About this task

An example of the target JCL for this scenario can be found in the product JCL library member CKZTRG. A detailed example follows.

Procedure

1. Specify the PARMLIB data set. Note that the DB2 Cloning Tool Table Space Cloning target job ignores fields it doesn't need. Those fields are TCPIP_OPTIONS and DSN_COPY_OPTIONS. This allows the same PARMLIB member to be used for the source, target and TCP/IP server jobs. See line #3 in the target job example and the associated note 3.
2. Specify the input data set where the DB2 Cloning Tool Table Space Cloning source job created input parameters for the target job. This can be found in the CKZSYNC DD statement in the source job example, line 7. See line #6 in the target job example and the associated note 6.
3. Specify the input data set where the DB2 Cloning Tool Table Space Cloning source job created SQL to update the sequence number for identity columns. This can be found in the CKZSQL DD statement in the source job example, line 9. See line #7 in the target job example and the associated note 7.

Target job example

For completeness and to illustrate where JCL parameters and DB2 Cloning Tool Table Space Cloning control statements must match, the following JCL includes sample DB2 Cloning Tool Table Space Cloning control statements.

The source job JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```
//jobcard
/*JOBPARM S=trgsys
/******
/*      COPYRIGHT 2009 MAINSTAR SOFTWARE CORPORATION      *
/*      ALL RIGHTS RESERVED                                *
/******
/*
/* Scenarios 1-4 - the target job is the same for all 4 scenarios.
/* The only change is the data set specified in CKZSQL and CKZIN.
/* These are output from the source job.
/*
/* The target job must be run on the target DB2 system.
/*
/* variables to be filled in ...
/*
/* jobcard - job card
/* trgsys - name of target system
/* hlq1 - CKZ LOADLIB and PARMLIB high level qualifier
/* hlq2 - SYSMDUMP high level qualifier
/* dsnvr1 - DB2 high level qualifier
/* parmbr - CKZ PARMLIB member
/* dumpsc - SYSMDUMP STORCLAS if needed
/* dumpmc - SYSMDUMP MGMTCLAS if needed
/* hlq3 - CKZ users high level qualifier
/* lstdmbr - LISTDEF member name
/*
/*
/* Initiate DFSMSdss to EXECUTE FLASHCOPY or SNAPSHOT
/*
1 //STEP1 EXEC PGM=CKZ00500,REGION=0M
2 //STEPLIB DD DISP=SHR,DSN=hlq1.SCKZLOAD
```

```

//          DD DISP=SHR,DSN=dsnvrl.SDSNEXIT
//          DD DISP=SHR,DSN=dsnvrl.SDSNLOAD
//*
//*
//* CKZINI - Reads the CKZ parmlib to get default settings
//*
3 //CKZINI  DD DISP=SHR,DSN=h1q1.SCKZPARM(parmmbr)
//*
//*
//* CKZPRINT - primary message output, required
//*
4 //CKZPRINT DD SYSOUT=*
//*
//*
//* CKZLOG - detailed message output, optional
//*
5 //CKZLOG  DD SYSOUT=*
//*
//*
//* dump DDs
//*
//*SYSUDUMP DD SYSOUT=*          Note: SYSDUMP preferred
//*SYSDUMP DD DSN=h1q2.SYSDUMP,DISP=(NEW,CATLG),
//*          STORCLAS=dumpscl,MGMTCLAS=dumpmc,
//*          SPACE=(CYL,(50,30),RLSE),
//*          DCB=(LRECL=4160,BLKSIZE=4160,RECFM=FB),UNIT=SYSDA
//*ABNLIGNR DD DUMMY            do not remove if using ABENDAID
//*
//*
//* CKZSQL - output ALTER TABLE SQL needed to synchronize
//*
//* identity column values between target and source.
6 //CKZSQL  DD DISP=SHR,DSN=h1q3.SQLOUT(1stdmbr)
//*
//*
//* CKZIN - input commands that will be used by the target
//* job to make the VSAM object(s) accessible on the target
//* DB2 subsystem. Created by the source job.
//*
7 //CKZIN   DD DISP=SHR,DSN=h1q3.SYNCDB2(1stdmbr)
//*
//******
//* sample control statements as would appear in SYNCDB2 member
//* Note: all lines have /* added in column 1 for JCL compatibility.
//******
//*
//*CKZIN   DD *
//* 06313 09:16:18.55 JOBNAME=JMX8SRC JOBID=J0040889 */
//*          SOURCE SUBSYSTEM=DB8G TARGET SUBSYSTEM=V81S */
//*
//*SET TRGJOB(Y) LSSID(trss) SQLDD(CKZSQL) SCANO(N)
//*
//*
//* TABLE SPACES - COPIED SUCCESSFULLY */
//*
//*
//*SYNCDB2 TARGET-SSID (V81S) -
//* TARGET-DSN (DSN081D.DSNDBC.ZTNDB.ZTNITSS1.I0001.A001) -
//* SPACE-TYPE (TS ) -
//* PAGE-SIZE (4) -
//* VCAT (DSN081D) -
//* STOGROUP (ZSG884) -
//* START-SPACE (Y) -
//* RESET-LOGRBA (Y) -
//* XLATE (DBID,X'0120,X'0113, -
//*          PSID,X'0002,X'0047, -
//*          TSOB,X'0001,X'0046, -

```

```

/**          TBOB,X'0003,X'0048, -
/**          TBOB,X'0008,X'004D, -
/**          TBOB,X'000D,X'0052)
/**
/**SYNCDB2 TARGET-SSID (V81S) -
/** TARGET-DSN (DSN081D.DSNDBC.ZTNDB.ZTN3TSN1.I0001.A001) -
/** SPACE-TYPE (TS ) -
/** PAGE-SIZE (4) -
/** VCAT (DSN081D) -
/** STOGROUP (ZSG884) -
/** START-SPACE (Y) -
/** RESET-LOGRBA (Y) -
/** XLATE (DBID,X'0120,X'0113, -
/**          PSID,X'0024,X'0013, -
/**          TSOB,X'0023,X'0012, -
/**          TBOB,X'0025,X'0014, -
/**          TBOB,X'002A,X'0019, -
/**          TBOB,X'002F,X'001E)
/**
/**SYNCDB2 TARGET-SSID (V81S) -
/** TARGET-DSN (DSN081D.DSNDBC.ZTNDB.ZTN5TSN1.I0001.A001) -
/** SPACE-TYPE (TP ) -
/** PAGE-SIZE (4) -
/** VCAT (DSN081D) -
/** STOGROUP (ZSG884) -
/** START-SPACE (Y) -
/** RESET-LOGRBA (Y) -
/** XLATE (DBID,X'0120,X'0113, -
/**          PSID,X'0046,X'0035, -
/**          TSOB,X'0045,X'0034, -
/**          TBOB,X'0047,X'0036)
/**
/**SYNCDB2 TARGET-SSID (V81S) -
/** TARGET-DSN (DSN081D.DSNDBC.ZTNDB.ZTN5TSN1.I0001.A002) -
/** SPACE-TYPE (TP ) -
/** PAGE-SIZE (4) -
/** VCAT (DSN081D) -
/** STOGROUP (ZSG884) -
/** START-SPACE (Y) -
/** RESET-LOGRBA (Y) -
/** XLATE (DBID,X'0120,X'0113, -
/**          PSID,X'0046,X'0035, -
/**          TSOB,X'0045,X'0034, -
/**          TBOB,X'0047,X'0036)

```

Note: In this example, the following:

1. Execution of DB2 Cloning Tool Table Space Cloning main program.
2. DB2 Cloning Tool LOAD library must be authorized.
3. DD for the CKZINI member of PARMLIB. The CKZINI member contains the program variables.
4. CKZPRINT - Displays CKZINI tokens, CKZIN control parameters, DB2 SQL execution status, and SYNCDB2 status and START DB2 command status for each data set processed.
5. CKZLOG - Displays the DB2 commands issued by DB2 Cloning Tool Table Space Cloning and responses/results of the commands and detailed information about each DB2 page access. Not required.
6. DD for CKZSQL - The CKZSQL data set contains the SQL commands used by the target job to synchronize identity column values between the target and source.

7. DD for CKZIN - The CKZIN data set is where the source job created the input parameters for the target job. The source job DD statement is CKZSYNC DD. The sample job has typical target job input from CKZIN commented out.

Setup for scenario 4: Other copy methodologies - some or all target table spaces and index spaces do not exist in DB2 catalog

Ideally, all target DB2 table spaces and index spaces will exist before the DB2 Cloning Tool Table Space Cloning source job is executed. However, if there is a narrow maintenance window for the copy and there is not enough time to create the target table spaces and index spaces within the window, then the following procedure is available.

You may also use this procedure (shown in the following table) for the table spaces and index spaces that are inadvertently missed. The DB2 Cloning Tool Table Space Cloning source job will process existing and nonexistent target table spaces and index spaces in the same run. The output to some data sets will vary depending on whether the table spaces and index spaces are missing on the target or not..

Table 32. DB2 Cloning Tool Table Space Cloning - Setting up when using other copy methods and some or all target objects do not exist

Setup steps for other copy methodologies when some or all target objects do not exist
"Step 1: Create target DB2 table spaces and index spaces in target DB2 catalog."
"Step 2: Set up source job (CKZ4SRC)" on page 212.
"Step 3 (Optional): Set up stop target job (CKZSTPT)" on page 217.
"Step 4 (Optional): Set up and execute stop source job (CKZSTPS)" on page 205.
"Step 5: Set up non-DB2 Cloning Tool Table Space Cloning job to execute data set copies outside DB2 Cloning Tool Table Space Cloning" on page 219. Use the list of source and target data sets provided by DB2 Cloning Tool Table Space Cloning.
"Step 6 (Optional): Set up and execute job to re-start source DB2 table spaces and index spaces (CKZSTRS)" on page 206.
"Step 7 (Optional): Create missing target table spaces and index spaces" on page 220 the DB2 Cloning Tool Table Space Cloning source job in Step 2 of this procedure found missing from the target.
"Step 8 (Optional): Set up job to make copied VSAM objects available to target DB2 (CKZFIX)" on page 221.
"Step 9: Set up for a re-run of the source job" on page 222. Re-running the source job will update the control parameters for the target job so that it can work on the newly created table spaces and index spaces.
"Step 10: Set up and execute target job (CKZTRG)" on page 222.

Step 1: Create target DB2 table spaces and index spaces in target DB2 catalog

When some or all target table spaces and index spaces do not exist, the first step when copying by data set using copy methodologies other than FlashCopy, SnapShot, or TimeFinder/Clone is to create the target objects in the target DB2 catalog.

DB2 associates DB2 table spaces and index spaces with specific VSAM data set names. For example:

```
VCAT.DSNDBD.DBTEST1.TESTTBL.I0001.A001
```

where:

Node 1: VCAT - the data set high level qualifier

Node 2: DSNDBD is the data portion. DSNDBC is the cluster portion

Node 3: DBTEST1 - the database name in the DB2 catalog

Node 4: TESTTBL - the table space or index space name in the DB2 catalog

Node 5: Either I0001 or J0001 - a switch indicator

Node 6: A001 to E094A - a partition number or DB2 extent number

The correctly named DB2 table spaces and index spaces must exist in the target DB2 catalog for DB2 to recognize and use the VSAM data sets cloned to the target by DB2 Cloning Tool Table Space Cloning.

When DB2 creates VSAM objects, it puts internal object identifiers inside of them. Those internal object identifiers are probably not the same on the target as the internal object identifiers in the VSAM data sets copied from the source. In particular, the database ID will almost never be the same; however, the other IDs are relative to the database and will frequently be the same. DB2 Cloning Tool Table Space Cloning will have to translate the internal identifiers from the source to those of the target to make the cloned VSAM data sets usable by the target DB2. Therefore you must create those target table spaces and index spaces if they do not exist.

Create as many of the following as are required by your application:

- Create the target stogroup(s)
- Create the target database(s)
- Create the target table space(s)
- Create the target table(s)
- Create the target index space(s)
- Create any target primary keys
- Create any target foreign keys
- Create any target alias(es), view(s), synonym(s), constraint(s), trigger(s), stored procedure(s), etc.

Note: DB2 Cloning Tool Table Space Cloning can be used to find target table spaces and index spaces that do not exist. Submit the source job with PGM(NONE) and copy parameter COPY-IF-NO-DB2-TARGET-OBJECTS(N). DB2 Cloning Tool Table Space Cloning will issue a warning message each time a source table space or index space cannot be paired with a target table space or index space.

Step 2: Set up source job (CKZ4SRC)

The source job is comprised of several commands and phases to control the selection of the DB2 table spaces and index spaces on the source subsystem and to make them accessible on the target subsystem.

About this task

An example of the source JCL for this scenario can be found in the product JCL library member CKZ4SRC. A detailed example is provided for illustration after the procedure steps. Each step contains a reference to the example.

Procedure

1. Identify the source table spaces and index spaces to DB2 Cloning Tool Table Space Cloning using the DB2 Cloning Tool Table Space Cloning LISTDEF command.
 - The LISTDEF control statements can be either be entered into an existing PDS member (RECFM=FB,LRECL=80) and read as input by using the CKZLSTDF DD statement. See line #6 in the source job example and the associated note 6.
 - Or, the LISTDEF control statements can be read in-stream by using CKZLSTDF DD. See line #6 in the source job example and the associated note 6.
2. Specify the output data set where DB2 Cloning Tool Table Space Cloning will build the input parameters for the target job. See line #7 in the source job example and the associated note 7.
3. Identify the target DB2 subsystem information and copy method in the COPY command.
 - Identify the target DB2 subsystem to CKZIN in the COPY command, keyword TARGET-DB2(SSID(*n*)) where *n* is the name of your target DB2 subsystem. See line #14 in the source job example and the associated note 14.
 - Identify the default VCAT and target DB2 subsystem to CKZIN in the COPY command, keyword DEFVCAT(*n*) where *n* is a high level qualifier for the target DB2 subsystem data sets used by the target job if one or more target table spaces and index spaces do not exist in the DB2 catalog. See line #14 in the source job example and the associated note 14.
 - Since using a copy methodology other than FlashCopy, SnapShot, or TimeFinder/Clone to copy table spaces and index spaces, specify keyword DATA-MOVER(PGM(NONE)) in the COPY command to indicate that the copy is to be done outside of DB2 Cloning Tool Table Space Cloning. See line #14 in the source job example and the associated note 14.

Source job example

For completeness and to illustrate where JCL parameters and DB2 Cloning Tool Table Space Cloning control statements must match, the following JCL includes sample DB2 Cloning Tool Table Space Cloning control statements.

The source job JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```
//jobcard
/*JOBPARM S=srcsys
/*
/******
/*      COPYRIGHT 2009 MAINSTAR SOFTWARE CORPORATION      *
/*      ALL RIGHTS RESERVED                                *
/******
/*
/** Scenario 4 - one or more source spaces do not exist on the target
/**                and copies are done outside of CKZ
/**
/**  variables to be filled in ...
/**
/**  jobcard - job card
/**  srcsys  - name of source system
/**  hlq1    - CKZ LOADLIB and PARMLIB high level qualifier
/**  hlq2    - SYSMDUMP high level qualifier
/**  dsnvr1  - DB2 high level qualifier
```

```

/**      parmbr  - CKZ PARMLIB member
/**      dumpsc  - SYSDUMP STORCLAS if needed
/**      dumpmc  - SYSDUMP MGMTCLAS if needed
/**      hlq3    - CKZ users high level qualifier
/**      lstdmbr - LISTDEF member name
/**      trss    - target DB2 subsystem
/**      dddd    - DD DISP value if needed (must enable)
/**      uuuu    - DD UNIT value if needed (must enable)
/**      vvvvvv  - DD VOLUME value if needed (must enable)
/**      srss    - source DB2 subsystem
/**      defsqlid - default SQLID applied to LISTDEF objects if needed
/**      tloc    - target subsystem if using DDF (delete if not needed)
/**      ipaddr  - target IP if using TCPIP (delete if not needed)
/**      uid     - user id if using DDF (delete if not needed)
/**      password - password if using DDF (delete if not needed)
/**      hlqtrgob - VCAT for target objects that do not exist
/**      ocs     - object creator on source DB2 (delete if not needed)
/**      oct     - object creator on target DB2 (delete if not needed)
/**
/**
/**      Initiate DFSMSdss to EXECUTE FLASHCOPY or SNAPSHOT
/**
1 //STEP1   EXEC PGM=CKZ00500,REGION=0M
2 //STEPLIB DD DISP=SHR,DSN=hlq1.SCKZLOAD
  //        DD DISP=SHR,DSN=dsnvr1.SDSNEXIT
  //        DD DISP=SHR,DSN=dsnvr1.SDSNLOAD
  /**
  /**
  /**      CKZINI - Reads the CKZ parmlib to get default settings
  /**
3 //CKZINI  DD DISP=SHR,DSN=hlq1.SCKZPARAM(parmbr)
  /**
  /**
  /**      CKZPRINT - primary message output, required
  /**
4 //CKZPRINT DD SYSOUT=*
  /**
  /**
  /**      CKZLOG - detailed message output, optional
  /**
5 //CKZLOG  DD SYSOUT=*
  /**
  /**
  /**      dump DDs
  /**
  /**      *SYSUDUMP DD SYSOUT=*          Note: SYSDUMP preferred
  /**      *SYSDUMP DD DSN=hlq2.SYSDUMP,DISP=(NEW,CATLG),
  /**              STORCLAS=dumpsc,MGMTCLAS=dumpmc,
  /**              SPACE=(CYL,(50,30),RLSE),
  /**              DCB=(LRECL=4160,BLKSIZE=4160,RECFM=FB),UNIT=SYSDA
  /**      *ABNLIGNR DD DUMMY              do not remove if using ABENDAID
  /**
  /**
  /**      CKZLSTDF - CKZ uses LISTDEF like commands with standard IBM syntax
  /**      to select the source spaces to be copied to the target.
  /**
6 //CKZLSTDF DD DISP=SHR,DSN=hlq3.LISTDEF(lstdmbr)
  /**
  /**
  /**      CKZSYNC - output commands that will be used by the target
  /**      job to make the VSAM object(s) accessible on the target
  /**      DB2 subsystem.
  /**
7 //CKZSYNC DD DISP=OLD,DSN=hlq3.SYNCDB2(lstdmbr)
  /**
  /**
  /**      CKZCOPY - output list to copy with from data set name and to

```

```

      /* data set name
      /*
8 //CKZCOPY DD DISP=OLD,DSN=h1q3.COPYDSNS(1stdmbr)
      /*
      /*
      /* CKZSQL - output ALTER TABLE SQL needed to synchronize
      /* identity column values between target and source.
      /*
9 //CKZSQL DD DISP=OLD,DSN=h1q3.SQLOUT(1stdmbr)
      /*
      /*
      /* CKZIDCAM - output IDCAMS deletes and renames for all data sets
      /* copied to F0001, rather than I0001 or J0001.
      /*
      /*CKZIDCAM DD DISP=OLD,DSN=h1q3.IDCAMS(1stdmbr)
      /*
      /*
      /* CKZSTPT - output DB2 stop commands for all target spaces
      /*
10 //CKZSTPT DD DISP=OLD,DSN=h1q3.CMDSSTPT(1stdmbr)
      /*
      /*
      /* CKZSTPS - output DB2 stop commands for all source spaces
      /*
11 //CKZSTPS DD DISP=OLD,DSN=h1q3.CMDSSTPS(1stdmbr)
      /*
      /*
      /* CKZSTRS - output DB2 stop commands for all source spaces
      /*
12 //CKZSTRS DD DISP=OLD,DSN=h1q3.CMDSSTRS(1stdmbr)
      /*
      /*
      /* DSS output DD if required, see HLQDDDF below
      /*
      /*MYOUTDD DD DISP=dddd,UNIT=uuuu,VOL=SER=(vvvvvv)
      /*
      /******
      /* sample control statements
      /******
      /*
      /*CKZIN DD *
      /*
      /*
      /* Source DB2 subsystem name and default SQLID if needed
      /*
      /*
13 SET LOCAL-SSID(srss) DEFAULT-SQLID(defsqlid)
      /*
      /*
      /* CKZ sends this DD name to ADRDSSU as output (enable if needed)
      /*
      /*
      /* HLQDDDF HLQNAME(trgh1q) DDNAME(MYOUTDD)
      /*
      /*
      /*
      /* The COPY command prepares the objects ID translation
      /* parameters for the target job and outputs commands to
      /* assist the manual copy process.
      /*
      /*
14 COPY TARGET-DB2(SSID(trss) LOC(tloc) -
      IP(ipaddr) -
      USR(uid) PASSWORD(password) -
      DEFVCAT(h1qtrgob) ) -
      DATASETS-TO-COPY-DDN(CKZCOPY) -
      SYNCDB2-DDN(CKZSYNC) -
      SQLOUT-DDN(CKZSQL) -
      IDCAMS-DDN(CKZIDCAM) -
      STOP-TARGET-DDN(CKZSTPT) -
      STOP-SOURCE-DDN(CKZSTPS) -

```

```

START-SOURCE-DDN(CKZSTRS)      -
SIM(N)                          -
DATA-MOVER( PGM(NONE) )        -
REPLACE-TARGET-DSN(Y)         -
COPY-IF-NO-DB2-TARGET-OBJECTS(Y) -
OBJECT-TRANSLATE(CREATOR,ocs,oct)-
AUTO-STOP-TARGET-SPACE(Y)     -
AUTO-START-TARGET-SPACE(Y)    -
RESET-LOGRBA(Y)                -

```

Note: In this example, the following:

1. Execution of DB2 Cloning Tool Table Space Cloning main program.
2. DB2 Cloning Tool LOAD library must be authorized.
3. DD for the CKZINI member of PARMLIB. The CKZINI member contains the program variables.
4. CKZPRINT displays CKZINI tokens, control parameters, data set names and associated DB2 table spaces and index spaces, DB2 start and stop space command status and DFSMSdss program ADRDSSU commands and status.
5. CKZLOG - Displays LISTDEF processing and DB2 commands issued by DB2 Cloning Tool Table Space Cloning and responses/results of the commands. (This DD is not required.)
6. CKZLSTDF - DB2 Cloning Tool Table Space Cloning uses LISTDEF-like commands with standard IBM syntax to select the source table spaces and index spaces to be copied to the target. LISTDEF control statements can either be entered into an existing PDS member (RECFM=FB,LRECL=80) and read as input using the CKZLSTDF DD statement, or can be read in-stream using CKZLSTDF DD *
7. DD for the CKZSYNC data set. CKZSYNC contains the parameters in an existing PDS member that will be used by the target job to make the VSAM object accessible on the target DB2 subsystem. The target job will be submitted by the user after the source job completes successfully.
8. DD for CKZCOPY - The CKZCOPY data sets contains a list of input and output data sets if the keyword for the COPY command is: DATA-MOVER(PGM(NONE)). The purpose of this listing is to assist in copying the selected VSAM objects outside of DB2 Cloning Tool Table Space Cloning
9. DD for the CKZSQL data set. CKZSQL contains the SQL commands in an existing PDS member that will be used by the target job to synchronize identity column values between the target and source.
10. DD for CMDSSSTPT - The CMDSSSTPT data set contains the parameters that will be used by the CKZSTPT job to stop the target table spaces and index spaces if the copy is done outside of DB2 Cloning Tool Table Space Cloning.
11. DD for CMDSSSTPS - The CMDSSSTPS data set contains the parameters that will be used by the CKZSTPS job to stop the source table spaces and index spaces if the copy is done outside of DB2 Cloning Tool Table Space Cloning.
12. DD for CMDSSSTRS - The CMDSSSTRS data set contains the parameters that will be used by the CKZSTRS job to start the source table spaces and index spaces if the copy is done outside of DB2 Cloning Tool Table Space Cloning.
13. The SET LOCAL command specifies the local DB2 subsystem for the source job and the TCP/IP server job. DEFAULT-SQLID is applied to tables and indexes in LISTDEF without an explicit creator.
14. COPY Command - The COPY command with DATA-MOVER(PGM(NONE)) assumes the copy of the data sets was done outside of DB2 Cloning Tool Table Space Cloning, and only prepares the object ID transaction parameters for the

target job and outputs DB2 start and stop space commands and IDCAMS delete/define commands into data sets if requested.

Step 3 (Optional): Set up stop target job (CKZSTPT)

The CKZSTPT job will help copy data sets outside of DB2 Cloning Tool Table Space Cloning.

When the COPY command keyword is DATA-MOVER(PGM(NONE)), DB2 Cloning Tool Table Space Cloning writes DB2 stop parameters for the existing target DB2 table spaces and index spaces to a data set pointed to by COPY keyword STOP-TARGET-DDN(?). Note that DB2 Cloning Tool Table Space Cloning can only generate STOP commands for existing table spaces and index spaces. Another CKZ4SRC may be submitted after the target table spaces and index spaces are created to output all the target STOP commands.

Sample JCL for the CKZSTPT job is in the product JCL library, member CKZSTPT. Set up the job to point to the same data set as STOP-TARGET-DDN.

The CKZSTPT job JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

//jobcard
1 /*JOBPARM S=trgsys
  /******
  /*          COPYRIGHT 2009 MAINSTAR SOFTWARE CORPORATION          *
  /*                      ALL RIGHTS RESERVED                      *
  /******
  /*
  /* Use to stop target objects if a copy method outside CKZ
  /* is used to copy data sets from the source to the target.
  /*
  /* variables to be filled in ...
  /*
  /* jobcard - job card
  /* trgsys - name of target system
  /* dsnvr1 - DB2 high level qualifier
  /* hlq3 - CKZ users high level qualifier
  /* lstdmbr - LISTDEF member name
  /*
  /*
  /******
  /* Stop the DB2 target table and index spaces          *
  /******
  2 //STOPT EXEC PGM=IKJEFT01
    //STEPLIB DD DISP=SHR,DSN=dsnvr1.SDSNLOAD
    //SYSPRINT DD SYSOUT=*
    //SYSTSPRT DD SYSOUT=*
    /*
    /*
    /* SYSTSIN DD was CKZSTPT DD in the source job.
    /*
  3 //SYSTSIN DD DISP=SHR,DSN=hlq3.CMDSSTPT(lstdmbr)
  4 /* DSN SYSTEM(trss)
    /* -STOP DATABASE(DBASEA) SPACE(TSPACEA)
    /* -STOP DATABASE(DBASEA) SPACE(INDEXA)
    /*

```

Note: In this example, the following:

1. JOBPARM where *trgsys* is the target z/OS system.
2. Execution of TSO monitor program.

3. SYSTSIN – Data set referenced by source job COPY command keyword STOP-TARGET-DDN
4. SYSTSIN – DSN command. *trss* is the target SSID. This is supplied by DB2 Cloning Tool Table Space Cloning in the output data set.

Step 4 (Optional): Set up stop source job (CKZSTPS)

The purpose of the CKZSTPS job is to facilitate copying data sets outside of DB2 Cloning Tool Table Space Cloning.

When the COPY command keyword is DATA-MOVER(PGM(NONE)), DB2 Cloning Tool Table Space Cloning writes DB2 stop parameters for the source DB2 table spaces and index spaces to a data set pointed to by COPY keyword STOP-SOURCE-DDN(ddname).

Sample JCL for the CKZSTPTS job is in the product JCL library, member CKZSTPTS. Set up the job to point to the same data set as STOP-SOURCE-DDN.

The CKZSTPTS job JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

//jobcard
1 /*JOBPARM S=srcsys
  /******
  /*      COPYRIGHT 2009 MAINSTAR SOFTWARE CORPORATION          *
  /*      ALL RIGHTS RESERVED                                   *
  /******
  /*
  /** Use to stop source spaces if a copy method outside CKZ
  /** is used to copy data sets from the source to the target.
  /**
  /** variables to be filled in ...
  /**
  /** jobcard - job card
  /** srcsys - name of source system
  /** dsnvr1 - DB2 high level qualifier
  /** hlq3 - CKZ users high level qualifier
  /** lstdmbr - LISTDEF member name
  /**
  /**
2 //STOPS EXEC PGM=IKJEFT01
  //STEPLIB DD DISP=SHR,DSN=dsnvr1.SDSNLOAD
  //SYSPRINT DD SYSOUT=*
  //SYSTSPRT DD SYSOUT=*
  /**
  /**
  /** SYSTSIN DD was CKZSTPS DD in the source job.
  /**
3 //SYSTSIN DD DISP=SHR,DSN=hlq3.CMDSSTPS(lstdmbr)
4 /** DSN SYSTEM(srss)
  /** -STOP DATABASE(DBASEA) SPACE(TSPACEA)
  /** -STOP DATABASE(DBASEA) SPACE(INDEXA)
  /**

```

Note: In this example, the following:

1. JOBPARM where *srcsys* is the source z/OS system.
2. Execution of TSO monitor program.
3. SYSTSIN – Data set referenced by source job COPY command keyword STOP-SOURCE-DDN

4. SYSTSIN – DSN command. *srss* is the source SSID. This is supplied by DB2 Cloning Tool Table Space Cloning in the output data set.

Step 5: Set up non-DB2 Cloning Tool Table Space Cloning job to execute data set copies outside DB2 Cloning Tool Table Space Cloning

Employ whatever procedures and syntax appropriate to your chosen replication tool.

If requested by the COPY parameter DATASETS-TO-COPY-DDN, DB2 Cloning Tool Table Space Cloning writes out a list of the source and target VSAM object pairs. The target data set names will have the correct high level qualifier and the correct switch (I/J) indicator if target table spaces and index spaces exist or F0001 if target table spaces and index spaces do not exist. In addition, only the qualifying data set pairs will be listed.

For table spaces and index spaces that do not exist, we recommend that the name(s) of the target DB2 VSAM data set(s) be something that DB2 will not recognize.

Note: After creating the DB2 table spaces and index spaces (either manually or using a tool), we recommend that you change the first character of the fifth node of the data set name (I0001 or J0001) from an “I” or “J” to an “F”. Note that in the DB2 Cloning Tool Table Space Cloning data set copy list, this name is created automatically for the missing targets. If this naming convention is used, then the supplied CKZFIX job will make the data sets available to the target DB2. CKZFIX is documented in topic “Step 4: Set up and execute optional FIX job (CKZFIX)” on page 194.

A sample output:

```
* 06330 11:56:36.79 JOBNAME=JMX8SRC JOBID=J0044593
* SOURCE SUBSYSTEM=DB8G TARGET SUBSYSTEM=V81S
*
*
* TABLE SPACE DATA SETS TO BE COPIED
*
SOURCE DATASET=DSN081C.DSNDBC.ZTNDB.ZTN1TSS1.I0001.A001
TARGET DATASET=DSN081D.DSNDBC.ZTNDB.ZTN1TSS1.I0001.A001
*
SOURCE DATASET=DSN081C.DSNDBC.ZTNDB.ZTN2TSSC.I0001.A001
TARGET DATASET=DSN081D.DSNDBC.ZTNDB.ZTN2TSSC.J0001.A001
*
SOURCE DATASET=DSN081C.DSNDBC.ZTNDB.ZTN3TSN1.I0001.A001
TARGET DATASET=DSN081D.DSNDBC.ZTNDB.ZTN3TSN1.J0001.A001
```

Step 6 (Optional): Set up job to re-start source DB2 table spaces and index spaces (CKZSTRS)

The purpose of the CKZSTRS job is to facilitate copying data sets outside of DB2 Cloning Tool Table Space Cloning.

When the COPY command keyword is DATA-MOVER(PGM(NONE)), DB2 Cloning Tool Table Space Cloning writes DB2 start parameters for the source DB2 table spaces and index spaces to a data set pointed to by COPY keyword START-SOURCE-DDN(?).

Sample JCL for the CKZSTRS job is in the product JCL library, member CKZSTRS. Set up the job to point to the same data set as START-SOURCE-DDN.

The CKZSTRS job JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

//jobcard
1 /*JOBPARM S=srcsys
  /******
  /*      COPYRIGHT 2009 MAINSTAR SOFTWARE CORPORATION      *
  /*      ALL RIGHTS RESERVED                                *
  /******
  /*
  /** Use to start source objects if a copy method outside CKZ
  /** is used to copy data sets from the source to the target.
  /**
  /** variables to be filled in ...
  /**
  /** jobcard - job card
  /** srcsys - name of source system
  /** dsnvr1 - DB2 high level qualifier
  /** hlq3 - CKZ users high level qualifier
  /** lstdmbr - LISTDEF member name
  /**
  /**
2 //STARTS EXEC PGM=IKJEFT01
  //STEPLIB DD DISP=SHR,DSN=dsnvr1.SDSNLOAD
  //SYSPRINT DD SYSOUT=*
  //SYSTSPRT DD SYSOUT=*
  /**
  /**
  /** SYSTSIN DD was CKZSTRS DD in the source job.
  /**
3 //SYSTSIN DD DISP=SHR,DSN=hlq3.CMDSSTRS(lstdmbr)
4 /* DSN SYSTEM(srss)
  /* -START DATABASE(DBASEA) SPACE(TSPACEA)
  /* -START DATABASE(DBASEA) SPACE(INDEXA)
  /**

```

Note: In this example, the following:

1. JOBPARM where *srcsys* is the source z/OS system.
2. Execution of TSO monitor program.
3. SYSTSIN – Data set referenced by source job COPY command keyword START-SOURCE-DDN
4. SYSTSIN – DSN command. *srss* is the source SSID. This is supplied by DB2 Cloning Tool Table Space Cloning in the output data set.

Step 7 (Optional): Create missing target table spaces and index spaces

As the target table spaces and index spaces did not exist when the DB2 Cloning Tool Table Space Cloning source in step 2 was run, there are no STOP commands for the newly created table spaces and index spaces. If needed, at this point another DB2 Cloning Tool Table Space Cloning source job can be run to output these stop commands.

Use PGM(NONE) and no stops will be issued to the existing source spaces. Additionally, delete the IDCAMS-DDN parameter from the source job. If this delete

is not performed, the correct IDCAMS-DDN data set will be overwritten. The next step will delete these data sets and rename the copied data sets from F0001 to I0001.

Step 8 (Optional): Set up job to make copied VSAM objects available to target DB2 (CKZFIX)

The purpose of the CKZFIX job is to facilitate making the data sets copied to the target subsystem available to the target DB2.

When DB2 Cloning Tool Table Space Cloning detects missing target DB2 table spaces and index spaces, the output data set names are handled differently and parameters are written to data sets which will stop the missing table spaces and index spaces (after the user creates them), delete the new VSAM objects and rename the VSAM objects copied by DB2 Cloning Tool Table Space Cloning. .

Sample JCL for the FIX job is in the product JCL library, member CKZFIX. Set up the job to point to the same data sets as STOP-TARGET-DDN and IDCAMS-DDN.

The CKZFIX job JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

//jobcard
1 /*JOBPARM S=trgsys
//*
//*****
//*      COPYRIGHT 2009 MAINSTAR SOFTWARE CORPORATION      *
//*                      ALL RIGHTS RESERVED                *
//*****
//*
//* Use when missing DB2 objects created after the copy.
//*
//* variables to be filled in ...
//*
//* jobcard - job card
//* trgsys - name of target system
//* dsnvr1 - DB2 high level qualifier
//* hlq3 - CKZ users high level qualifier
//* lstdmbr - LISTDEF member name
//*
//*
//*****
//* Stop the DB2 created table space(s) AND index(es) *
//* for the deletion and rename step to follow.      *
//*****
//*
2 //STOP EXEC PGM=IKJEFT01
//*
//*
//STEPLIB DD DISP=SHR,DSN=dsnvr1.SDSNLOAD
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//*
//*
//* SYSTSIN DD was CKZSTPT DD in the source job CKZ2SRC
//*
3 //SYSTSIN DD DISP=SHR,DSN=hlq3.CMDSSTPT(lstdmbr)
4 /* DSN SYSTEM(trss)
//* -STOP DATABASE(DBASEA) SPACE(TSPACEA)
//* -STOP DATABASE(DBASEA) SPACE(INDEXA)
//*
//*
//*****

```

```

        /* Delete the DB2 created VSAM data set(s) and          *
        /*   rename the cloned data set(s)                      *
        /******
        /*
5 //DELNAME EXEC PGM=IDCAMS,COND=(4,LT)
  //SYSPRINT DD SYSOUT=*
  /*
  /*
  /* SYSIN DD was CKZIDCAM DD in the source job CKZ2SRC
  /*
6 //SYSIN   DD DISP=SHR,DSN=h1q3.IDCAMS(1stdmbr)
  /*
  /* DELETE DSN081D.DSNDBC.DBASEA.TSPACEA.I0001.A001 -
  /*   CLUSTER NOERASE PURGE
  /* ALTER -
  /*   DSN081D.DSNDBC.DBASEA.TSPACEA.F0001.A001 -
  /*   NEWNAME (DSN081D.DSNDBC.DBASEA.TSPACEA.I0001.A001)
  /* ALTER -
  /*   DSN081D.DSNDBD.DBASEA.TSPACEA.F0001.A001 -
  /*   NEWNAME (DSN081D.DSNDBD.DBASEA.TSPACEA.I0001.A001)
  /*
  /*
  /*
  /*

```

Note: In this example, the following:

1. JOBPARM where *trgsys* is the target z/OS system.
2. Execution of TSO monitor program.
3. SYSTSIN – Data set referenced by source job COPY command keyword STOP-TARGET-DDN.
4. SYSTSIN – DSN command. *trss* is the target SSID. This is supplied by DB2 Cloning Tool Table Space Cloning in the output data set.
5. Execution of IDCAMS – To delete the VSAM objects created by the target DB2 when the table spaces and index spaces were created and rename the VSAM objects copied from the source DB2.
6. SYSIN – Data set referenced by source job COPY command keyword IDCAMS-DDN.

Step 9: Set up for a re-run of the source job

Re-running the source job at this point will capture the object IDs of the newly created DB2 table spaces and index spaces and update the data sets used as input for the target job..

The COPY command keyword must be DATA-MOVER(PGM(NONE)).

Step 10: Set up and execute target job (CKZTRG)

The target job is comprised of a single step and SYNCDB2 command to make the VSAM objects accessible on the target subsystem. It will optionally stop and start the target table spaces and index spaces.

About this task

An example of the target JCL for this scenario can be found in the product JCL library member CKZTRG. A detailed example follows.

Procedure

1. Specify the PARMLIB data set. Note that the DB2 Cloning Tool Table Space Cloning target job ignores fields it doesn't need. Those fields are TCPIP_OPTIONS and DSN_COPY_OPTIONS. This allows the same PARMLIB member to be used for the source, target and TCP/IP server jobs. See line #3 in the target job example and the associated note 3.
2. Specify the input data set where the DB2 Cloning Tool Table Space Cloning source job created input parameters for the target job. This can be found in the CKZSYNC DD statement in the source job example, line 7. See line #6 in the target job example and the associated note 6.
3. Specify the input data set where the DB2 Cloning Tool Table Space Cloning source job created SQL to update the sequence number for identity columns. This can be found in the CKZSQL DD statement in the source job example, line 9. See line #7 in the target job example and the associated note 7.

Target job example

For completeness and to illustrate where JCL parameters and DB2 Cloning Tool Table Space Cloning control statements must match, the following JCL includes sample DB2 Cloning Tool Table Space Cloning control statements.

The source job JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```
//jobcard
/*JOBPARM S=trgsys
/******
/*      COPYRIGHT 2009 MAINSTAR SOFTWARE CORPORATION      *
/*              ALL RIGHTS RESERVED                      *
/******
/*
/* Scenarios 1-4 - the target job is the same for all 4 scenarios.
/* The only change is the data set specified in CKZSQL and CKZIN.
/* These are output from the source job.
/*
/* The target job must be run on the target DB2 system.
/*
/* variables to be filled in ...
/*
/* jobcard - job card
/* trgsys - name of target system
/* hlq1 - CKZ LOADLIB and PARMLIB high level qualifier
/* hlq2 - SYSMDUMP high level qualifier
/* dsnvr1 - DB2 high level qualifier
/* parmbr - CKZ PARMLIB member
/* dumpsc - SYSMDUMP STORCLAS if needed
/* dumpmc - SYSMDUMP MGMTCLAS if needed
/* hlq3 - CKZ users high level qualifier
/* lstdmbr - LISTDEF member name
/*
/*
/* Initiate DFSMSdss to EXECUTE FLASHCOPY or SNAPSHOT
/*
1 //STEP1 EXEC PGM=CKZ00500,REGION=0M
2 //STEPLIB DD DISP=SHR,DSN=hlq1.SCKZLOAD
// DD DISP=SHR,DSN=dsnvr1.SDSNEXIT
// DD DISP=SHR,DSN=dsnvr1.SDSNLOAD
/*
/*
/* CKZINI - Reads the CKZ parmlib to get default settings
/*
```

```

3 //CKZINI DD DISP=SHR,DSN=h1q1.SCKZPARM(parmmbr)
  /**
  /**
  /** CKZPRINT - primary message output, required
  /**
4 //CKZPRINT DD SYSOUT=*
  /**
  /**
  /** CKZLOG - detailed message output, optional
  /**
5 //CKZLOG DD SYSOUT=*
  /**
  /**
  /** dump DDs
  /**
  /***SYSUDUMP DD SYSOUT=* Note: SYSDUMP preferred
  /***SYSDUMP DD DSN=h1q2.SYSDUMP,DISP=(NEW,CATLG),
  /** STORCLAS=dumpscl,MGMTCLAS=dumpscl,
  /** SPACE=(CYL,(50,30),RLSE),
  /** DCB=(LRECL=4160,BLKSIZE=4160,RECFM=FB),UNIT=SYSDA
  /***ABNLIGNR DD DUMMY do not remove if using ABENDAID
  /**
  /**
  /** CKZSQL - output ALTER TABLE SQL needed to synchronize
  /**
  /** identity column values between target and source.
6 //CKZSQL DD DISP=SHR,DSN=h1q3.SQLOUT(1stdmbr)
  /**
  /**
  /** CKZIN - input commands that will be used by the target
  /** job to make the VSAM object(s) accessible on the target
  /** DB2 subsystem. Created by the source job.
  /**
7 //CKZIN DD DISP=SHR,DSN=h1q3.SYNCDB2(1stdmbr)
  /**
  /*******
  /** sample control statements as would appear in SYNCDB2 member
  /** Note: all lines have /** added in column 1 for JCL compatibility.
  /*******
  /**
  /***CKZIN DD *
  /** 06313 09:16:18.55 JOBNAME=JMX8SRC JOBID=J0040889 */
  /** SOURCE SUBSYSTEM=DB8G TARGET SUBSYSTEM=V81S */
  /**
  /***SET TRGJOB(Y) LSSID(trss) SQLDD(CKZSQL) SCANO(N)
  /**
  /**
  /** TABLE SPACES - COPIED SUCCESSFULLY */
  /**
  /**
  /***SYNCDB2 TARGET-SSID (V81S) -
  /** TARGET-DSN (DSN081D.DSNDBC.ZTNDB.ZTN1TSS1.I0001.A001) -
  /** SPACE-TYPE (TS ) -
  /** PAGE-SIZE (4) -
  /** VCAT (DSN081D) -
  /** STOGROUP (ZSG884) -
  /** START-SPACE (Y) -
  /** RESET-LOGRBA (Y) -
  /** XLATE (DBID,X'0120,X'0113, -
  /** PSID,X'0002,X'0047, -
  /** TSOB,X'0001,X'0046, -
  /** TBOB,X'0003,X'0048, -
  /** TBOB,X'0008,X'004D, -
  /** TBOB,X'000D,X'0052)
  /**
  /***SYNCDB2 TARGET-SSID (V81S) -
  /** TARGET-DSN (DSN081D.DSNDBC.ZTNDB.ZTN3TSN1.I0001.A001) -

```

```

/** SPACE-TYPE (TS ) -
/** PAGE-SIZE (4) -
/** VCAT (DSN081D) -
/** STOGROUP (ZSG884) -
/** START-SPACE (Y) -
/** RESET-LOGRBA (Y) -
/** XLATE (DBID,X'0120,X'0113, -
/**         PSID,X'0024,X'0013, -
/**         TSOB,X'0023,X'0012, -
/**         TBOB,X'0025,X'0014, -
/**         TBOB,X'002A,X'0019, -
/**         TBOB,X'002F,X'001E)
/**
/**SYNCDB2 TARGET-SSID (V81S) -
/** TARGET-DSN (DSN081D.DSNDBC.ZTNDB.ZTN5TSN1.I0001.A001) -
/** SPACE-TYPE (TP ) -
/** PAGE-SIZE (4) -
/** VCAT (DSN081D) -
/** STOGROUP (ZSG884) -
/** START-SPACE (Y) -
/** RESET-LOGRBA (Y) -
/** XLATE (DBID,X'0120,X'0113, -
/**         PSID,X'0046,X'0035, -
/**         TSOB,X'0045,X'0034, -
/**         TBOB,X'0047,X'0036)
/**
/**SYNCDB2 TARGET-SSID (V81S) -
/** TARGET-DSN (DSN081D.DSNDBC.ZTNDB.ZTN5TSN1.I0001.A002) -
/** SPACE-TYPE (TP ) -
/** PAGE-SIZE (4) -
/** VCAT (DSN081D) -
/** STOGROUP (ZSG884) -
/** START-SPACE (Y) -
/** RESET-LOGRBA (Y) -
/** XLATE (DBID,X'0120,X'0113, -
/**         PSID,X'0046,X'0035, -
/**         TSOB,X'0045,X'0034, -
/**         TBOB,X'0047,X'0036)

```

Note: In this example, the following:

1. Execution of DB2 Cloning Tool Table Space Cloning main program.
2. DB2 Cloning Tool LOAD library must be authorized.
3. DD for the CKZINI member of PARMLIB. The CKZINI member contains the program variables.
4. CKZPRINT - Displays CKZINI tokens, CKZIN control parameters, DB2 SQL execution status, and SYNCDB2 status and START DB2 command status for each data set processed.
5. CKZLOG - Displays the DB2 commands issued by DB2 Cloning Tool Table Space Cloning and responses/results of the commands and detailed information about each DB2 page access. Not required.
6. DD for CKZSQL - The CKZSQL data set contains the SQL commands used by the target job to synchronize identity column values between the target and source.
7. DD for CKZIN - The CKZIN data set is where the source job created the input parameters for the target job. The source job DD statement is CKZSYNC DD. The sample job has typical target job input from CKZIN commented out.

Chapter 16. Procedures for cloning DB2 table spaces and index spaces

This topic describes different methods for copying table spaces and index spaces from one DB2 subsystem to another.

The procedures described here provide for various situations, such as the data set copy methodology used to copy the table spaces and index spaces, and whether the table spaces and index spaces already exist in the target DB2 catalog.

Choose the appropriate procedure for your situation.

Copy procedure 1: FlashCopy, SnapShot, or TimeFinder/Clone setup - when all target table spaces and index spaces already exist in DB2 catalog

Use the procedure shown in the following table if you want to copy the source table spaces and index spaces to the target DB2 subsystem when the target table spaces and index spaces already exist in the DB2 catalog and you are using FlashCopy, SnapShot, or TimeFinder/Clone.

Table 33. DB2 Cloning Tool Table Space Cloning - copy procedure for FlashCopy, SnapShot, or TimeFinder/Clone when objects exist in the DB2 catalog

Copy procedure steps
"Step 1: Verify/create target DB2 table spaces and index spaces in target DB2 catalog."
"Step 2 (Optional): Submit TCP/IP server job." If the source and target DB2 subsystems are on different LPARs, and either DDF is unavailable or target table spaces and index spaces must be stopped by the source job, then submit the optional TCP/IP server job CKZTCPS.
"Step 3: Submit source job" on page 228.
"Step 4: Submit target job" on page 228.

Step 1: Verify/create target DB2 table spaces and index spaces in target DB2 catalog

Ensure that the target DB2 table spaces and index spaces exist in the target DB2 catalog.

Do this manually, use a non-DB2 Cloning Tool Table Space Cloning tool, or execute the DB2 Cloning Tool Table Space Cloning source job. When using the DB2 Cloning Tool Table Space Cloning source job, specify COPY option DATA-MOVER(PGM(NONE) and COPY-IF-NO-DB2-TARGET-OBJECTS(N). DB2 Cloning Tool Table Space Cloning will output a warning message for each target table space and index space not found.

Step 2 (Optional): Submit TCP/IP server job

The TCP/IP server job is only necessary if the source and target DB2 subsystems are on different LPARs, and either DDF is unavailable or target table spaces and index spaces must be stopped by the source job.

DB2 Cloning Tool Table Space Cloning uses DDF instead of TCP/IP to read the target catalog; however, TCP/IP is required to issue cross system DB2 STOP and DISPLAY commands. Target DB2 catalog access is necessary to confirm the existence of the target DB2 table spaces and index spaces, compare attributes between source and target table spaces and index spaces, and retrieve information from the target DB2 catalog for ID translations.

Step 3: Submit source job

The next step is to submit the source job.

The source job:

- Connects to the source and target DB2 subsystems
- Selects the table spaces and index spaces to be replicated from the source DB2 catalog
- Confirms the existence and compatibility of the target table spaces and index spaces
- Stops the source and target table spaces and index spaces
- Invokes FlashCopy, SnapShot, or TimeFinder/Clone (if available) to replicate the data sets
- Starts the source table spaces and index spaces
- Prepares the parameters needed by the target job to complete the cloning process

Step 4: Submit target job

The next step is to submit the target job.

The target job must run on the same z/OS system as the target DB2. The target job uses the parameters passed to it by the source job to optionally restart the target DB2 table spaces and index spaces and update the internal identifiers in the copied VSAM objects to make them accessible to DB2.

This completes the cloning process.

Copy procedure 2: FlashCopy, SnapShot, or TimeFinder/Clone setup - when some or all target table spaces and index spaces do not exist in DB2 catalog

Ideally, all target DB2 table spaces and index spaces will exist before the DB2 Cloning Tool Table Space Cloning source job is executed. However, if there is a narrow maintenance window for the copy, and there is not enough time to fit creating the target table spaces and index spaces within the window, then the following procedure is available.

You may also use this procedure (shown in the following table) for the table spaces and index spaces that are inadvertently missed. The DB2 Cloning Tool Table Space Cloning source job will process existing and nonexistent target table spaces and index spaces in the same run. The output to some data sets will vary depending on whether the table spaces and index spaces are missing on the target or not.

Table 34. DB2 Cloning Tool Table Space Cloning - copy procedure for FlashCopy, SnapShot, or TimeFinder/Clone when some or all target objects do not exist in the DB2 catalog

Copy procedure steps
“Step 1: Verify/create target DB2 table spaces and index spaces in target DB2 catalog” on page 227.
“Step 2 (Optional): Submit TCP/IP server job” on page 227. If the source and target DB2 subsystems are on different LPARs, and either DDF is unavailable or target table spaces and index spaces must be stopped by the source job, then submit the optional TCP/IP server job CKZTCPS.
“Step 3: Submit source job.”
“Step 4: Create missing target DB2 table spaces and index spaces - manually or using non-DB2 Cloning Tool Table Space Cloning Tool” on page 230.
“Step 5: Submit FIX job on target z/OS system (CKZFIX)” on page 230.
“Step 6: Rerun source job” on page 231. Ensure that the copy command keyword is DATA-MOVER(PGM(NONE)).
“Step 7: Submit target job” on page 231.

Step 1: Verify/create target DB2 table spaces and index spaces in target DB2 catalog

Ensure that the target DB2 table spaces and index spaces exist in the target DB2 catalog.

Do this manually, use a non-DB2 Cloning Tool Table Space Cloning tool, or execute the DB2 Cloning Tool Table Space Cloning source job. When using the DB2 Cloning Tool Table Space Cloning source job, specify COPY option DATA-MOVER(PGM(NONE)) and COPY-IF-NO-DB2-TARGET-OBJECTS(N). DB2 Cloning Tool Table Space Cloning will output a warning message for each target table space and index space not found.

Step 2 (Optional): Submit TCP/IP server job

The TCP/IP server job is only necessary if the source and target DB2 subsystems are on different LPARs, and either DDF is unavailable or target table spaces and index spaces must be stopped by the source job.

DB2 Cloning Tool Table Space Cloning uses DDF instead of TCP/IP to read the target catalog; however, TCP/IP is required to issue cross system DB2 STOP and DISPLAY commands. Target DB2 catalog access is necessary to confirm the existence of the target DB2 table spaces and index spaces, compare attributes between source and target table spaces and index spaces, and retrieve information from the target DB2 catalog for ID translations.

Step 3: Submit source job

The next step is to submit the source job.

The source job:

- Connects to the source and target DB2 subsystems
- Selects the table spaces and index spaces to be cloned from the source DB2 catalog

- Confirms the existence and compatibility of the target table spaces and index spaces
- Stops the source and existing target table spaces and index spaces
- Invokes FlashCopy, SnapShot, or TimeFinder/Clone (if available) to replicate the data sets
- Starts the source table spaces and index spaces
- Prepares the parameters needed by the target job to complete the cloning process

If the source job detects missing DB2 table spaces or index spaces on the target DB2, DB2 Cloning Tool Table Space Cloning will:

- Use as the high level qualifier, of the data set name(s) that it creates, the alias supplied by the DEFVCAT keyword of the COPY command
- Modify the target data set name(s) that DB2 Cloning Tool Table Space Cloning creates. The first character of the fifth node will be changed from "I" or "J" to "F". DB2 will not recognize the new data set name; therefore, the data set cannot be mistakenly corrupted when the missing DB2 table spaces or index spaces are created.

Note: Ordinarily, DB2 dynamically allocates a VSAM data set when a new table space or index space is created. However, if DB2 finds a correctly named data set already in existence, DB2 will use that data set and write internal object IDs into it to associate it with the table space or index space that DB2 created. If data already exists in the VSAM data set, then that data may become permanently inaccessible. New data can be inserted into the data set but the old data may not be retrieved.

- Generate IDCAMS parameters to delete the VSAM object(s) which will, later, be created by the target DB2
- Generate IDCAMS parameters to rename the "F" data sets to a name recognizable to DB2.

Step 4: Create missing target DB2 table spaces and index spaces - manually or using non-DB2 Cloning Tool Table Space Cloning Tool

Once the data set copy is done, you can create the missing table spaces, tables, and index spaces for the target DB2. This step provides the missing object IDs DB2 Cloning Tool Table Space Cloning needs to clone the VSAM objects.

The execution of the source job in Step 3 of this procedure cannot write out STOP commands for non-existent target table spaces and index spaces. If STOP commands are required, run another source job at this point with PGM(NONE) to generate STOP commands for those missing target table spaces and index spaces. These stop commands can then be used in "Step 5: Submit FIX job on target z/OS system (CKZFIX)."

Step 5: Submit FIX job on target z/OS system (CKZFIX)

The CKZFIX job uses DB2 commands and IDCAMS parameters passed to it by the source job.

The CKZFIX job will:

- Stop the target DB2 table spaces and index spaces (if available)
- Delete the new VSAM object(s) created by DB2

- Rename the copied VSAM objects to the name(s) of the deleted VSAM object(s)

Ordinarily, DB2 dynamically allocates a VSAM data set when a new table space or index space is created. However, if DB2 finds a correctly named data set already in existence, DB2 will use that data set and write internal object IDs into it to associate it with the table space or index space that DB2 created. If data already exists in the VSAM data set, then that data becomes permanently inaccessible. New data can be inserted into the data set but the old data cannot be retrieved.

Making the target data sets unrecognizable to DB2 prevents it from writing the internal identifiers for any newly created DB2 table spaces and index spaces into those data sets. The target DB2 will allocate new data sets and write into them. The new and empty data sets can then be replaced with data sets copied from the source DB2.

Step 6: Rerun source job

During the re-run of the source job, DB2 Cloning Tool Table Space Cloning will update the parameters for the target job to include the new object IDs.

Step 7: Submit target job

The next step is to submit the target job.

The target job must run on the same z/OS system as the target DB2. The target job uses the parameters passed to it by the source job to optionally restart the target DB2 table spaces and index spaces and update the internal identifiers in the copied VSAM objects to make them accessible to DB2.

This completes the cloning process.

Copy procedure 3: All other copy methodologies - All target table spaces and index spaces already exist in DB2 catalog

Use the procedure shown in the table if you want to copy the source table spaces and index spaces to the target DB2 subsystem when the target table spaces and index spaces already exist in the DB2 catalog and you are using a method other than FlashCopy, SnapShot, or TimeFinder/Clone.

Table 35. DB2 Cloning Tool Table Space Cloning - copy procedure for other copy methods when all target objects already exist in the DB2 catalog

Copy procedure steps
"Step 1: Verify/create target DB2 table spaces and index spaces in target DB2 catalog" on page 227.
"Step 2 (Optional): Submit TCP/IP server job" on page 227. If the source and target DB2 subsystems are on different LPARs, and either DDF is unavailable or target table spaces and index spaces must be stopped by the source job, then submit the optional TCP/IP server job CKZTCPS.
"Step 3: Submit source job" on page 232 to capture the names of the data sets to be copied. It will also create DB2 commands to stop the source and target table spaces and index spaces and to re-start the source table spaces and index spaces.
"Step 4: Submit job CKZSTPT to stop target DB2 table spaces and index spaces" on page 233.
"Step 5: Submit job CKZSTPS to stop source DB2 table spaces and index spaces" on page 233.

Table 35. DB2 Cloning Tool Table Space Cloning - copy procedure for other copy methods when all target objects already exist in the DB2 catalog (continued)

Copy procedure steps
"Step 6: Submit non-DB2 Cloning Tool Table Space Cloning job to copy VSAM objects" on page 233.
"Step 7: Submit job CKZSTRS to start source DB2 table spaces and index spaces" on page 233.
"Step 8: Rerun source job" on page 233.
"Step 9: Submit target job" on page 233.

Step 1: Verify/create target DB2 table spaces and index spaces in target DB2 catalog

Ensure that the target DB2 table spaces and index spaces exist in the target DB2 catalog.

Do this manually, use a non-DB2 Cloning Tool Table Space Cloning tool, or execute the DB2 Cloning Tool Table Space Cloning source job. When using the DB2 Cloning Tool Table Space Cloning source job, specify COPY option DATA-MOVER(PGM(NONE)) and COPY-IF-NO-DB2-TARGET-OBJECTS(N). DB2 Cloning Tool Table Space Cloning will output a warning message for each target table space and index space not found.

Step 2 (Optional): Submit TCP/IP server job

The TCP/IP server job is only necessary if the source and target DB2 subsystems are on different LPARs, and either DDF is unavailable or target table spaces and index spaces must be stopped by the source job.

DB2 Cloning Tool Table Space Cloning uses DDF instead of TCP/IP to read the target catalog; however, TCP/IP is required to issue cross system DB2 STOP and DISPLAY commands. Target DB2 catalog access is necessary to confirm the existence of the target DB2 table spaces and index spaces, compare attributes between source and target table spaces and index spaces, and retrieve information from the target DB2 catalog for ID translations.

Step 3: Submit source job

The next step is to submit the source job.

When the COPY command keyword is DATA-MOVER(PGM(NONE)) it is assumed that the copy will be done outside of DB2 Cloning Tool Table Space Cloning.

DB2 Cloning Tool Table Space Cloning still determines the source and target DB2 subsystems, selects from the source DB2 catalog the table spaces and index spaces to be cloned, and confirms the existence of the target table spaces and index spaces. It also checks for object incompatibilities between source and target table spaces and index spaces.

To assist the user, DB2 Cloning Tool Table Space Cloning writes the list of source and target data set names provided by the LISTDEF command to a data set. It also writes stop commands for the source and target DB2 table spaces and index spaces and writes start commands for the source table spaces and index spaces to data sets. This information can be used to prepare the non-DB2 Cloning Tool Table Space Cloning copy job.

Step 4: Submit job CKZSTPT to stop target DB2 table spaces and index spaces

Existing target table spaces and index spaces must be stopped so that DB2 will deallocate the VSAM objects. This allows them to be replaced.

Step 5: Submit job CKZSTPS to stop source DB2 table spaces and index spaces

The source table spaces and index spaces must be stopped so that DB2 will deallocate the VSAM objects. This allows them to be copied.

Step 6: Submit non-DB2 Cloning Tool Table Space Cloning job to copy VSAM objects

Any hardware or software replication utility can be used.

Step 7: Submit job CKZSTRS to start source DB2 table spaces and index spaces

Once the data sets are replicated (copied) the source table spaces and index spaces are no longer needed by the cloning process.

Step 8: Rerun source job

If you skipped the running the source job in step 3 of this procedure, then this will actually be the first time that the source job is run.

The source job:

- Connects to the source and target DB2 subsystems
- Selects the table spaces and index spaces to be replicated from the source DB2 catalog
- Confirms the existence and compatibility of the target table spaces and index spaces
- Prepares the parameters needed by the target job to complete the cloning process

Step 9: Submit target job

The next step is to submit the target job.

The target job must run on the same z/OS system as the target DB2. The target job uses the parameters passed to it by the source job to optionally restart the target DB2 table spaces and index spaces and update the internal identifiers in the copied VSAM objects to make them accessible to DB2. This completes the cloning process.

Copy procedure 4: All other copy methodologies - some or all target table spaces and index spaces do not exist in DB2 catalog

Ideally, all target DB2 table spaces and index spaces will exist before the DB2 Cloning Tool Table Space Cloning source job is executed. However, if there is a narrow maintenance window for the copy, and there is not enough time to fit creating the target table spaces and index spaces within the window, then the following procedure is available. You may also use this procedure for the table spaces and index spaces that are inadvertently missed.

The DB2 Cloning Tool Table Space Cloning source job will process existing and nonexistent target table spaces and index spaces in the same run. The output to some data sets will vary depending on whether the table spaces and index spaces are missing on the target or not.

The following table lists the copy procedure when some or all target objects do not exist in the DB2 catalog:

Table 36. DB2 Cloning Tool Table Space Cloning - copy procedure for other copy methods when some or all target objects do not exist in the DB2 catalog

Copy procedure steps
"Step 1: Verify/create target DB2 table spaces and index spaces in target DB2 catalog" on page 227.
"Step 2 (Optional): Submit TCP/IP server job" on page 227. If the source and target DB2 subsystems are on different LPARs, and either DDF is unavailable or target table spaces and index spaces must be stopped by the source job, then submit the optional TCP/IP server job CKZTCPS.
"Step 3: Submit source job" on page 235.
"Step 4: Create missing target DB2 table spaces and index spaces" on page 236.
"Step 5: Submit job CKZSTPT to stop target DB2 table spaces and index spaces" on page 236.
"Step 6: Submit job CKZSTPS to stop source DB2 table spaces and index spaces" on page 236.
"Step 7: Submit non-DB2 Cloning Tool Table Space Cloning job to copy VSAM objects" on page 236.
"Step 8: Submit job CKZSTRS to restart source DB2 table spaces and index spaces" on page 236.
"Step 9: Submit FIX job on target z/OS system (CKZFIX)" on page 236.
"Step 10: Rerun the source job to prepare target job" on page 237.
"Step 11: Submit target job" on page 237.

Step 1: Verify/create target DB2 table spaces and index spaces in target DB2 catalog

Ensure that the target DB2 table spaces and index spaces exist in the target DB2 catalog.

Do this manually, use a non-DB2 Cloning Tool Table Space Cloning tool, or execute the DB2 Cloning Tool Table Space Cloning source job. When using the DB2 Cloning Tool Table Space Cloning source job, specify COPY option DATA-MOVER(PGM(NONE) and COPY-IF-NO-DB2-TARGET-OBJECTS(N). DB2 Cloning Tool Table Space Cloning will output a warning message for each target table space and index space not found.

Step 2 (Optional): Submit TCP/IP server job

The TCP/IP server job is only necessary if the source and target DB2 subsystems are on different LPARs, and either DDF is unavailable or target table spaces and index spaces must be stopped by the source job.

DB2 Cloning Tool Table Space Cloning uses DDF instead of TCP/IP to read the target catalog; however, TCP/IP is required to issue cross system DB2 STOP and DISPLAY commands. Target DB2 catalog access is necessary to confirm the

existence of the target DB2 table spaces and index spaces, compare attributes between source and target table spaces and index spaces, and retrieve information from the target DB2 catalog for ID translations.

Step 3: Submit source job

DB2 Cloning Tool Table Space Cloning can facilitate using methodologies other than FlashCopy, SnapShot, or TimeFinder/Clone to replicate the DB2 data sets. If you are interested in using the information provided by DB2 Cloning Tool Table Space Cloning for this purpose, then submit the source job at this point. The COPY command keyword must be DATA-MOVER(PGM(NONE)). If you are not interested in the information that DB2 Cloning Tool Table Space Cloning can provide, then skip this step. IBM does not recommend skipping this step.

When the COPY command keyword is DATA-MOVER(PGM(NONE)), it is assumed that the copy will be done outside of DB2 Cloning Tool Table Space Cloning. DB2 Cloning Tool Table Space Cloning still determines the source and target DB2 subsystems, selects from the source DB2 catalog the table spaces and index spaces to be cloned, and confirms the existence of the target objects.

To assist the user, DB2 Cloning Tool Table Space Cloning writes the list of source and target data set names provided by the LISTDEF command to a data set. It also writes stop commands for the source and target DB2 table spaces and index spaces, and restart commands for the source table spaces and index spaces, to data sets. This information can be used to prepare the non-DB2 Cloning Tool Table Space Cloning copy job.

If the source job detects missing DB2 table spaces and index spaces on the target DB2, it will:

- Use the alias supplied by the DEFVCAT keyword of the COPY command as the high level qualifier of the data set name(s) that it creates.
- Modify the target data set name(s) that it derives. The first character of the fifth node will be changed from "I" or "J" to "F". If this naming convention is adhered to, then DB2 will not recognize the new data set name; therefore, the data set cannot be mistakenly corrupted when the missing DB2 table spaces and index spaces are created. Furthermore, DB2 Cloning Tool Table Space Cloning will create parameters to assist in renaming the data sets.

Note: Ordinarily, DB2 dynamically allocates a VSAM data set when a new table space or index space is created. However, if it finds a correctly named data set already in existence, it will use that data set and write internal object IDs into it to associate it with the table space or index space that DB2 created. If data already exists in the VSAM data set, then that data becomes permanently inaccessible. New data can be inserted into the data set but the old data cannot be retrieved.

- Generate IDCAMS parameters to delete the VSAM object(s) which will, later, be created by the target DB2,
- Generate IDCAMS parameters to rename the "F" data sets to a name recognizable to DB2.

Step 4: Create missing target DB2 table spaces and index spaces

Once the missing table spaces and index spaces are flagged, you can create the missing table spaces, tables and indexes for the target DB2. This step provides the missing object IDs DB2 Cloning Tool Table Space Cloning needs to make the VSAM objects accessible.

Step 5: Submit job CKZSTPT to stop target DB2 table spaces and index spaces

Existing target table spaces and index spaces must be stopped so that DB2 will deallocate the VSAM objects. This allows them to be replaced.

This step is optional if the copy utility does not require the source VSAM objects to be deallocated and a FUZZY-COPY is acceptable.

Step 6: Submit job CKZSTPS to stop source DB2 table spaces and index spaces

The source table spaces and index spaces must be stopped so that DB2 will deallocate the VSAM objects. This allows them to be copied.

Step 7: Submit non-DB2 Cloning Tool Table Space Cloning job to copy VSAM objects

Any hardware or software replication utility can be used.

Step 8: Submit job CKZSTRS to restart source DB2 table spaces and index spaces

Once the data sets are replicated (copied) the source table spaces and index spaces are no longer needed by the cloning process.

You may start them manually via DDL or use the start commands provided by the tool. This step is not necessary if the source table spaces and index spaces were not stopped.

Step 9: Submit FIX job on target z/OS system (CKZFIX)

The CKZFIX job uses DB2 commands and IDCAMS parameters passed to it by the source job.

The CKZFIX job will:

- Delete the new VSAM object(s) created by DB2
- Rename the copied VSAM objects to the name(s) of the deleted VSAM object(s)

Ordinarily, DB2 dynamically allocates a VSAM data set when a new table space or index space is created. However, if DB2 finds a correctly named data set already in existence, DB2 will use that data set and write internal object IDs into it to associate it with the table space or index space that DB2 created. If data already exists in the VSAM data set, then that data becomes permanently inaccessible. New data can be inserted into the data set but the old data cannot be retrieved.

Making the target data sets unrecognizable to DB2 prevents it from writing the internal identifiers for any newly created DB2 table spaces and index spaces into

those data sets. The target DB2 will allocate new data sets and write into them. The new and empty data sets can then be replaced with data sets copied from the source DB2.

Step 10: Rerun the source job to prepare target job

If you skipped the running the source job in step 3 of this procedure, then this will actually be the first time that the source job is run.

The source job:

- Connects to the source and target DB2 subsystems
- Selects the table spaces and index spaces to be replicated from the source DB2 catalog
- Confirms the existence and compatibility of the target table spaces and index spaces
- Prepares the parameters needed by the target job to complete the cloning process

Step 11: Submit target job

The next step is to submit the target job.

The target job must run on the same z/OS system as the target DB2. The target job uses the parameters passed to it by the source job to optionally restart the target DB2 table spaces and index spaces and update the internal identifiers in the copied VSAM objects to make them accessible to DB2.

This completes the cloning process.

Chapter 17. Using data masking with table space cloning

Data copied from a source object(s) to a target object(s) may be modified during the copy so that the target data in one or more columns may be different from the source data. The changes are made based on masking rules that are enabled during the copy. Examples of fields that you might change are social security numbers, credit card numbers, names and addresses. In addition, DB2 Cloning Tool Table Space Cloning can generate index rebuild jobs for the changed objects as part of the cloning process.

If the copy is not successful or the target object does not exist, then no masking is performed.

You can apply data masking by using the batch interface commands and DDs. Data masking is also available using the ISPF interface.

Summary of steps

1. Add keywords to the COPY command.
2. Add a DD to the source job.
3. Create the masking command members.
4. Copy and update a job template member for the job to rebuild the indexes.
5. Add DDs to the target job for the index rebuilds.
6. Submit the source job.
7. Submit the target job.
8. Submit the job to rebuild the indexes.

Types of data masking

DB2 Cloning Tool Table Space Cloning offers the following types of data masking:

- **STATIC:** masks column values with a constant value.
- **MASK:** modifies values by replacing positions within an existing column value with a customized pattern or static value.
- **PATTERN:** modifies values by replacing an existing column value with a pattern; this type allows you to specify the pattern based on the existing value of the column.
- **RANDOM:** masks values using random numeric or date/timestamp data, optionally within a certain range.
- **USEREXIT:** allows the user to specify their own user exit containing data masking rules.
- **SEQUENCE:** masks by generating a sequence of numeric values with a specified increment.
- **SCRAMBLE:** applies a static scrambling algorithm to the column value.
- **CURRENT DATE:** replaces the column values with the current date.
- **CURRENT TIME:** replaces the column values with the current time.
- **CURRENT TIMESTAMP:** replaces the column values with the current timestamp.
- **CURRENT USER:** replaces the column values with the current user ID of the owner of the process when it is run.

Objects and data types not supported

Data masking is not supported with compressed table spaces.

The following data types may not be masked using DB2 Cloning Tool Table Space Cloning data masking:

- All graphic types (GRAPHIC, VARGRAPHIC, LONGVARG)
- All LOB types (BLOB, CLOB, DBCLOB)
- XML

Masking for tables with RI

When using data masking on tables with referential integrity, the only types of supporting masking functions are SCRAMBLE, PATTERN, and USEREXIT. All referentially related columns will have the data mask applied, even if the masking function is only specified for one column. In addition, a user exit applied to a parent or foreign key column must generate repeatable results.

Source job changes

Make the following changes to your source job in order to use data masking.

Add keywords to the COPY command

Add the DATA-MASKING, INCLUDE-ALL-RI, and TARGET-JOB-INDEX-REBUILD-DDN keywords to the COPY command.

Create a member containing MASKDEF commands and keywords

The member is to contain MASKDEF commands that identify the table and column to be masked and define the masking rule. The MASKDEF commands must be formatted as follows. Multiple MASKDEF commands can be included in the member.

```
MASKDEF
  TABLENAME(tbname)
  TABLECREATOR(tbcreator)
  COLUMNNAME(colname)
  MASKRULE(maskrule)
```

TABLENAME (tbname)

(Required) Specify the name of the source table that contains the column to be masked.

TABLECREATOR(tbcreator)

Specify the table creator of the source table containing the column to be masked. The DEFAULT-SQLID is used if this keyword is not included.

COLUMNNAME(colname)

(Required) Specify the column name (in the source table) to be masked.

MASKRULE(maskrule)

(Required) Include the mask rule for the column. Refer to “Specifying the masking rule” on page 241 for syntax and detailed information about the mask rule.

Add a new DD for the MASKDEF data set

Add a DD pointing to the MASKDEF member as follows. This DD is required if DATA-MASKING(Y).

```
CKZMSKDF DD DISP=SHR,DSN=h1q.MASKDEF(mbrname)
```

Updating the template for the job to rebuild indexes

Make a copy of either the CKZJOBI (recommended) or CKZJOBR template in the SCKZJCL library and modify it for your site.

This template contains the statements used to build the REBUILD INDEX utility job. Instructions for updating the template are contained in the member.

Specifying the masking rule

DB2 Cloning Tool Table Space Cloning applies data masks to columns based on the masking rule that you specify in the MASKDEF command.

To prevent errors during data masking, you must know the specific data type of the column that is being masked and ensure that the DB2 Cloning Tool Table Space Cloning can process that data type.

Continuation rules

A rule that requires continuation must only use one continuation character. For example:

```
RULE(STATIC, "long_string_3456789B123456789C123456789D123456789E1-  
23456789F123456789G123456789H123456789I123456789J123456789K12345678")
```

More than one continuation character is not allowed, such as:

```
RULE(STATIC, "long_string_3456789B123456789C12-  
3456789D123456789E1-  
23456789F123456789G123456789H123456789I123456789J123456789K12345678")
```

MASKRULE(STATIC, *numeric* | "date_or_time" | "string")

This mask rule defines a constant value to be used.

The value must be placed between quotation marks for DATE, TIME, TIMESTAMP, CHAR, and VARCHAR data types. For NUMERIC data types (SMALLINT, INTEGER, BIGINT, REAL, DOUBLE, DECFLOAT and DECIMAL), the value must be entered without quotation marks. The keyword STATIC can be defined in upper or lower case.

This rule can be used for the following column data types:

- SMALLINT
- INTEGER
- BIGINT
- REAL
- DOUBLE
- DECFLOAT
- DECIMAL
- DATE

- TIME
- TIMESTAMP
- TIMESTAMP WITH TIMEZONE (DB2 10 and later)
- CHAR
- VARCHAR

Parameters

numeric

This value can be any of the following :

- A binary integer (small integer, integer, or big integer)
- A decimal number
- A floating point number (real, double, or decimal)

date_or_time

This value can be a time, date, timestamp; for DB2 10 and later, it can be a timestamp with time zone.

string

This value can be any alphanumeric string.

Samples

```
MASKRULE(STATIC, 10)
MASKRULE(STATIC, -100)
MASKRULE(STATIC, -7.2e+75)
MASKRULE(STATIC, "2009-02-04")
MASKRULE(STATIC, "2008-12-01-15.30.30")
MASKRULE(STATIC, "2010-10-31-23.59.59.000000000000+10:00")
MASKRULE(STATIC, "123 Division Street")
```

MASKRULE(MASK, “*pattern*”, *start*, *end*)

This mask rule modifies values by replacing positions within an existing value with the specified pattern or static value.

Non-alphanumeric characters can be generated by enclosing them with a backslash (\) when they are specified. For example, \% generates %. The keyword MASK can be defined in upper or lower case.

This rule can be used for the following column data types:

- CHAR
- VARCHAR

Parameters

pattern

Use this variable to specify the value to be placed in the position specified in the start and end fields. You can also type a pattern to be evaluated, then placed in the specified position or positions. Patterns can be specified in three ways: character generation, string selector, or static value.

- Character generation: Characters that are enclosed in square brackets force a random selection of one of the enclosed characters. Ranges of characters can be established by using the dash. Ranges can only include alphanumeric characters. Quantification after a symbol or group of symbols determines how many times this expression can be repeated, as shown in the following table:

Table 37. Declaration examples

Declaration	Repeat count
{n}	Exactly n
{m,n}	From m till n inclusive

In combination with repeated templates, a set of characters can establish a correspondence with real text, such as digit columns, phone numbers, zip codes, HTML page elements, and so on.

A set of possible symbols must be defined in brackets. For example, [abc] allows one of those three characters to appear in the text. [1234567890] allows any of those digits to be used, as shown in the following table:

Table 38. Character generation examples

Character specification	Evaluates to
[Aa]	A or a
[abcde] or [a-e]	Any character between a and e, inclusive
[0-9]	Any single-digit number
[0-9a-z]	Any single-digit number or any lowercase letter
[A-Z]{3}	Any three-character uppercase string
[24\%\-\&]	One of the following: 2, 4, %, -, or &
[AaBb]{5}	Any five-character string containing any combination of the four specified letters. For example: AAbaA or abbBA or AbAbB
[Aab2]{1,17}	A string from one to 17 characters in length made up of any combination of the four specified letters. For example: a2baab2A or A or a2baabAbaA22bAb2A

- String selector: Strings that are enclosed in parentheses and delimited by the pipe (|) character perform a random selection of one of the strings. Character generators can be included in a string selector. Both alphanumeric and non-alphanumeric characters can be generated. Non-alphanumeric characters can be generated by dereferencing them (using a \) when they are specified. The following table shows some string selector examples.

Table 39. String selector examples

Character specification	Evaluates to
(Mrs Mr Ms)	Mrs, Mr, or Ms
(C[ATO] A[KLR])	CA, CT, CO, AK, AL, or AR

- Static Value: Any string of characters that is not enclosed in brackets (for character generation) or parentheses (for string selection), is considered a static value and is concatenated in the order of appearance. A non-alphanumeric character should be preceded by a backslash (\), as shown in the following table:

Table 40. Static value examples

Character specification	Evaluates to
(Sir The\ Great Mr) Bill	Sir Bill, The Great Bill, or Mr Bill

start, end

Type an inclusive range of the positions in the source data that are to be replaced.

Samples

```

MASKRULE(MASK, "[abc]{1}", 1, 2)
MASKRULE(MASK, "[0]{1}", 1, 2)
MASKRULE(MASK, "[0-9]{10}", 1, 10)
MASKRULE(MASK, "[A-Z]{10}", 1, 10)
MASKRULE(MASK, "[a-z]{5}", 1, 5)
MASKRULE(MASK, "[ABCabc123]{1, 10}", 1, 10)
MASKRULE(MASK, "[\^\`!\@#\$\%\&\*\(\)\_\-=\+]{10}", 1, 10)
MASKRULE(MASK, "[A-Za-z0-9]{1, 10}", 1, 10)
MASKRULE(MASK, "[A-Za-z0-9\^\`!\@#\$\%\&\*\(\)\_\-=\+]{1, 10}", 1, 10)
MASKRULE(MASK, "(Mrs|Mr|Ms)", 1, 3)
MASKRULE(MASK, "(C[ATO]|A[KLK])", 1, 3)
MASKRULE(MASK, "(Sir|The\ Great|Mr) Bill", 1, 10)
MASKRULE(MASK, "(Sir|The\ Great|Mr) Bill (C[ATO]|A[A-Za-z0-9])", 1, 20)
MASKRULE(MASK, "[Aab2]", {1,17})
MASKRULE(MASK, "(IL|IA|IN)")
MASKRULE(MASK, "(I[L|A|N]")

```

MASKRULE(PATTERN, “pattern”, “use_sources”)

The pattern rule allows you to specify a pattern to be used to generate a value. A variety of patterns can be generated by specifying a formula for the pattern.

The keyword PATTERN can be defined in upper or lower case.

This rule can be used for the following column data types:

- CHAR
- VARCHAR

Parameters**pattern**

This parameter defines the pattern that will generate a value. Nested expressions can be used. All other characters are directly inserted into the rule value. There are several different pattern types that can be created. Each type of pattern is created by typing the appropriate pattern in the pattern field.

- Character generation: Characters that are enclosed in square brackets force a random selection of one of the enclosed characters. Ranges of characters can be established by using the dash. Ranges can only include alphanumeric characters. Quantification after a symbol or group of symbols determines how many times this expression can be repeated.

Declaration	Repeat count
{n}	Exactly n
{m,n}	From m till n inclusive

In combination with repeated templates, a set of characters can establish a correspondence with real text, such as digit columns, phone numbers, zip codes, HTML page elements, and so on.

A set of possible symbols must be defined in brackets. For example, [abc] allows one of those three characters to appear in the text. [1234567890] allows any of those digits to be used.

Table 41. Character generation examples

Character specification	Evaluates to
[Aa]	A or a
[abcde] or [a-e]	Any character between a and e, inclusive
[0-9]	Any single-digit number
[0-9a-z]	Any single-digit number or any lowercase letter
[A-Z]{3}	Any three-character uppercase string
[24\%\-\&]	One of the following: 2, 4, %, -, or &
[AaBb]{5}	Any five-character string containing any combination of the four specified letters. For example: AAbaA or abbBA or AbAbB
[Aab2]{1,17}	A string from one to 17 characters in length made up of any combination of the four specified letters. For example: a2baab2A or A or a2baabAbaA22bAb2A

- String selector: Strings that are enclosed in parentheses and delimited by the pipe (|) character perform a random selection of one of the strings. Character generators can be included in a string selector. Both alphanumeric and non-alphanumeric characters can be generated. Non-alphanumeric characters can be generated by dereferencing them (using a \) when they are specified.

Table 42. String selector examples

Character specification	Evaluates to
(Mrs Mr Ms)	Mrs, Mr, or Ms
(C[ATO] A[KLR])	CA, CT, CO, AK, AL, or AR

- Static Value: Any string of characters that is not enclosed in brackets (for character generation) or parentheses (for string selection), is considered a static value and is concatenated in the order of appearance. A non-alphanumeric character should be preceded by a backslash (\).

Table 43. Static value examples

Character specification	Evaluates to
(Sir The\ Great Mr) Bill	Sir Bill, The Great Bill, or Mr Bill

use_sources

Valid values for this option are Y(ES) or N(O). If this option is Y(es), output data will be generated based on the current value of the field. This function is based on static formula; for any given input value and identical pattern, the output value will be the same. For example, for the pattern:

```
[0-2][0-9]{3}\-[0-9]{4}\-[A-D]{4}
```

and the field is 5985-9597-BDHF, the output result will be always be 2896-1648-ABCD.

Samples

```
MASKRULE(PATTERN, "[0-2][0-9]{3}\-[0-9]{4}\-[A-D]{4}", "Y")
MASKRULE(PATTERN, "[0-2][0-9]{3}\-[0-9]{4}\-[A-D]{4}", Y)
MASKRULE(PATTERN, "[abc]{1}", YES)
```

```

MASKRULE(PATTERN, "[0]{1}", Y)
MASKRULE(PATTERN, "[0-9]{10}", NO)
MASKRULE(PATTERN, "[A-Z]{10}", N)
MASKRULE(PATTERN, "[a-z]{5}", YES)
MASKRULE(PATTERN, "[ABCabc123]{1, 10}", Y)
MASKRULE(PATTERN, "[\~\`!\@#\$\%\&\*\(\)\_\-\=\+]{10}", NO)
MASKRULE(PATTERN, "[A-Za-z0-9]{1, 10}", N)
MASKRULE(PATTERN, "[A-Za-z0-9\~\`!\@#\$\%\&\*\(\)\_\-\=\+]{1, 10}", YES)
MASKRULE(PATTERN, "(Mrs|Mr|Ms)", Y)
MASKRULE(PATTERN, "(C[AT0]|A[KLR])", NO)
MASKRULE(PATTERN, "(Sir|The\ Great|Mr) Bill", N)
MASKRULE(PATTERN, "(Sir|The\ Great|Mr) Bill (C[AT0]|A[A-Za-z0-9])", YES)

```

MASKRULE(RANDOM, “min”, “max”, “check_bounds”)

MASKRULE(RANDOM, "min_timestamp", "max_timestamp", "min_timezone", "max_timezone", "check_bounds")

The random rule produces random numeric or date/timestamp data within certain bounds.

The keyword RANDOM can be defined in upper or lower case.

This rule can be used for the following column data types:

- SMALLINT
- INTEGER
- BIGINT
- DECIMAL
- DATE
- TIME
- TIMESTAMP
- TIMESTAMP WITH TIMEZONE (DB2 10 and later)

Parameters

min, max

min and max must be entered between quotation marks for DATE, TIME and TIMESTAMP data types. All numeric values must be entered without quotation marks. These values can be any of the following, but both min and max must be defined and must be the same data type:

- A binary integer (small integer, integer, or big integer)
- A decimal number
- A floating point number (real or double)
- A time, date, or a timestamp

Minimum and maximum values must be specified in the same data type format of the table column data type. For example, if the column type where the mask is to be applied is decimal, then minimum and maximum should be defined in format NNNNN.MM, where precision and scale also correspond to the column data type

If min and max do not correspond to the column data type, DB2 Cloning Tool Table Space Cloning returns an error.

For *min*, enter the lowest possible value to be generated by this rule. For *max*, enter the highest possible value to be generated by this rule.

min_timestampz, max_timestampz, min_timezone, max_timezone

min_timestampz and max_timestampz are used to calculate minimum and maximum time in UTC. min_timestampz after conversion to UTC must be less than max_timestampz after conversion to UTC.

min_timezone and max_timezone will be used to calculate a random TIMEZONE for the final random value TIMESTAMP WITH TIMEZONE. These parameters are optional; if not included, the system time zone of the target system will be used.

After calculation, random value boundaries (min_timestampz (UTC) + min_timezone) and (max_timestampz (UTC) + max_timezone) must be in the range "0001-01-01-00.00.00.000000000000+00:00" to "9999-12-31-23.59.59.999999999999+00:00" inclusive. Otherwise, a range error message will be displayed.

check_bounds

Valid values for this parameter are Y(ES) or N(O). Enter Y or YES to have one row in your target table that corresponds to the value specified in the min field and one row in your table that corresponds to the value specified in the max field. This option allows you to test the endpoints of the range that you specified.

Samples

```
MASKRULE(RANDOM, 0, 1, N)
MASKRULE(RANDOM, -1, 0, NO)
MASKRULE(RANDOM, -100, 0, Y)
MASKRULE(RANDOM, -1, 1, YES)
MASKRULE(RANDOM, 0.00, 100.00, NO)
MASKRULE(RANDOM, -100.00, 0.00, Y)
MASKRULE(RANDOM, "2008-01-01", "2008-12-31", Y)
MASKRULE(RANDOM, "2008-12-01", "2009-01-31", N)
MASKRULE(RANDOM, "00.00.00", "23.59.59", Y)
MASKRULE(RANDOM, "12.00.00", "12.59.59", N)
MASKRULE(RANDOM, "2008-01-01-00.00.00", "2008-12-31-23.59.59.999999", Y)
MASKRULE(RANDOM, "2008-12-01-12.00.00", "2009-01-31-12.59.59.999999", N)
MASKRULE(RANDOM, "2008-01-01-00.00.00+10:00",
"2008-12-31-23.59.59.9999999999-10:00", "-05:00", "+12:30", Y)
```

MASKRULE(USEREXIT, *module*)

The user exit rule allows DB2 Cloning Tool Table Space Cloning to call an exit that defines the user's own data masking rules.

The keyword USEREXIT can be defined in upper or lower case.

This rule can be used for any column data type.

Parameters

module

For *module*, enter the external module name that will be called for each column value. The current value of the column will be used as input for the module. The user exit should change the value and return it to DB2 Cloning Tool Table Space Cloning in the same buffer. The user exit must reside in the STEPLIB concatenation.

The length of the column values cannot be changed using a user exit.

The following format is used to pass the column value to the user exit:

```
int user_exit(void* field_data, size_t length);
```

Samples

`MASKRULE(USEREXIT, CSNMASK1)`

MASKRULE(SEQUENCE, *initial_value*, *increment*)

The sequence rule generates a sequence of numeric values.

The first time the function reference is evaluated, it returns the value of the first argument (*initial_value*). Each subsequent evaluation returns the value of the second argument (*increment*) added to the previously returned value. The resulting data type is always integer.

The keyword `SEQUENCE` can be defined in upper or lower case.

This rule can be used for the following column data types:

- `SMALLINT`
- `INTEGER`
- `BIGINT`

Parameters

initial_value

Enter an integer value that will be the first value in the sequence.

increment

Enter an integer value that will be increment each value in the sequence.

Samples

```
MASKRULE(SEQUENCE, 2789, 2)
MASKRULE(SEQUENCE, 100, 10)
MASKRULE(SEQUENCE, -100, 10)
```

MASKRULE(SCRAMBLE)

The scramble rule applies a predefined DB2 Cloning Tool Table Space Cloning scrambling function to the value specified by the argument. The result has the same data type as the provided argument.

The keyword `SCRAMBLE` can be defined in upper or lower case.

This rule can be used for the following column data types:

- `SMALLINT`
- `INTEGER`
- `BIGINT`
- `REAL`
- `DOUBLE`
- `DECFLOAT`
- `DECIMAL`
- `DATE`
- `TIME`
- `TIMESTAMP`
- `CHAR`
- `VARCHAR`
- `BINARY`

- VARBINARY

Parameters

None.

Samples

`MASKRULE(SCRAMBLE)`

MASKRULE(CURRENT DATE)

MASKRULE(CURRENT TIME)

MASKRULE(CURRENT TIMESTAMP, "timezone")

These date and time-related rules replace the column values with the current date, current time or current timestamp values.

The keywords `CURRENT DATE`, `CURRENT TIME`, and `CURRENT TIMESTAMP` can be defined in upper or lower case.

This rule can be used for the following column data types:

- DATE
- TIME
- TIMESTAMP
- TIMESTAMP WITH TIMEZONE (DB2 10 and later)

Parameters

timezone

For DB2 10 and later, `timezone` can be specified with `CURRENT TIMESTAMP`. Valid values are from -12:59 to +14:00. If the value is omitted, then the system time zone of the target system will be used.

Samples

```
MASKRULE(CURRENT DATE)
MASKRULE(CURRENT TIME)
MASKRULE(CURRENT TIMESTAMP)
MASKRULE(CURRENT TIMESTAMP, "-04:00")
```

MASKRULE(CURRENT USER)

The current user rule replaces the column value with the current user ID of the owner of the process when it is run.

The keyword `CURRENT USER` can be defined in upper or lower case.

This rule can be used for the following column data types:

- CHAR
- VARCHAR

Parameters

None.

Samples

```
MASKRULE(CURRENT USER)
```

Target job changes

After the source job is executed, the target job will contain the MASKCMD command added to the SYNCDB2 data set. MASKCMD includes all the information from the source job MASKDEF command (read from the source job maskdef DD), as well as the data set name and whether all partitions are included.

Add DDs to the target job for the index rebuild job

Add two DDs to the target job:

```
//ddnameI DD DISP=SHR,DSN=hlq.indsn(mbr)
```

where *ddname* is the DD specified in the TARGET-JOB-INDEX-REBUILD keyword. You must append an I to *ddname*. *hlq.indsn(mbr)* is the data set location and member name of the template you edited.

```
//ddnameO DD DISP=OLD,DSN=hlq.outdsn(mbr)
```

where *ddname* is the DD specified in the TARGET-JOB-INDEX-REBUILD keyword. You must append an O to *ddname*. *hlq.outdsn(mbr)* is the data set location and member name where you want the utility job to be saved. This DD will contain the output from the generated REBUILD INDEX job.

MASKCMD example

After the source job is executed, the target job will contain the MASKCMD command. An example follows:

```
MASKCMD TABLENAME(tbname)
        TABLECREATOR(tbcreator)
        COLNAME(colname)
        MASKRULE(maskrule)
        DSN(dsn)
        ALLPARTS(Y|N)
```

In addition, there is one TABLEDEF command for each table with a masked column in a successfully copied data set (or data sets, if partitioned). For tables with RI, there must be at least one PARENT and CHILD subcommand of TABLEDEF for each column with an RI relationship. For example:

```
TABLEDEF
  (TBCREATOR(creator)           -
   TBNAME(name)                 -
   DBNAME(name)                 -   DB table resides in
   TSNAME(name)                 -   TS table resides in
   CCSID(char)                  -   encoding scheme of the table
   OBID(x'n)                     -   target table object ID
   REL(Y|N)                      -   if Y, rels exist for this table
   COLNO(X'number of columns)    -   # of columns in this table
   PARENTNO(X'number of parents) -   # of parents
   CHILDNO(X'number of children) -   # of children
   COLUMN(name,                  column name
    coltype,                     column type (CHAR, INT, etc.)
    colno,                       number of column in this table
    length,                      column length
    scale,                       scale if decimal column
    nulls,                       Y or N (nulls exist)
    ccsid                        column CCSID in hex

  {,PARENT,
    creator -
  ,table -
  ,relname -
```

```

        ,# of columns -
        ,column -
    . . . ,column_n -
    . . .
        ,creator -
        ,table -
        ,relname -
        ,# of columns -
        ,column -
    . . . ,column_n -
    . . .
    )}
    {,CHILD,
        creator -
        ,table -
        ,relname -
        ,# of columns -
        ,column -
    . . . ,column_n -
    . . .
        ,creator -
        ,table -
        ,relname -
        ,# of columns -
        ,column -
    . . . ,column_n -
    . . .
    )}

```

When data masking is specified, DB2 Cloning Tool Table Space Cloning also adds a DATA-MASKING (Y) keyword to the SET command syntax in the SYNCDB2 member.

Submitting the job to rebuild indexes

When the target job is executed, the job to rebuild indexes is generated and placed in the *ddnameO* output data set specified in the target job.

Submit the job to rebuild indexes.

Examples of masking jobs

This topic shows sample masking members.

Sample MASKDEF member (input to source job)

```

MASKDEF -
  TBCR(PDDONAA) -
  TBNM(Z@63TB11) -
  CLNM(COL1Z@63TB11CHAR) -
  RULE(STATIC,"3458-08-06")

MASKDEF -
  TBCR(PDDONAA) -
  TBNM(Z@63TB11) -
  CLNM(COL2Z@63TB11CHAR2) -
  RULE(MASK,"YA-Za-z0-9"{1,10}", 1, 10)

MASKDEF -
  TBCR(PDDONAA) -
  TBNM(Z@63TB11) -
  CLNM(COL6Z@63TB11INT) -
  RULE(RANDOM, 1, 100, YES)

```

```

MASKDEF
  TBCR(PDDONAA)
  TBNM(Z@63TB11)
  CLNM(COL5Z@63TB11SMALLI)
  RULE(SEQUENCE, 1, 1)

```

Sample SYNCDB2 member after execution of the source job

```

/* 09054 11:40:38.35 JOBNAME=PDDONAX8 JOBID=J0831512 */
/* SOURCE SUBSYSTEM=D8A TARGET SUBSYSTEM=T81B */

```

```

SET TRGJOB(Y) LSSID(T81B) SQLDD(CKZQT81B)
SCANO(N)
MAX-SUBTASKS(1)
DATA-MASKING(Y)

```

```

/* TABLESPACES - COPIED SUCCESSFULLY */

```

SYNCDB2 commands

. . .

```

MASKCMD -
  TBCR (PDDONAA) -
  TBNM (Z@63TB11) -
  CLNM (COL1Z@63TB11CHAR) -
  RULE (STATIC,"3458-08-06") -
  DSN (CKZTRG8.DSNDBC.Z@63DB.Z@63TS1.I0001.A001) -
  ALLP (Y)

```

```

TABLEDEF -
  TBCREATOR (PDDONAA) -
  TBNAME (Z@63TB11) -
  DBNAME (Z@63DB) -
  TSNAME (Z@63TS1) -
  CCSID (E) -
  OBID (X'0003) -
  REL (N) -
  COLNO (X'0006) -
  PARENTNO (X'0000) -
  CHILDNO (X'0001) -
  COLUMN (
    COL1Z@63TB11CHAR -
    ,CHAR -
    ,X'0001 -
    ,X'0028 -
    ,X'0000 -
    ,N -
    ,X'00000025 -
    COL2Z@63TB11CHAR2 -
    ,CHAR -
    ,X'0002 -
    ,X'000A -
    ,X'0000 -
    ,N -
    ,X'00000025 -
    COL3Z@63TB11IDENT -
    ,INTEGER -
    ,X'0003 -
    ,X'0004 -
    ,X'0000 -
    ,N -
    ,X'00000000 -
    COL4Z@63TB11ROWID -
    ,ROWID -
  )

```



```

,X'0004 -
,X'0011 -
,X'0000 -
,N -
,X'00000000 -
,COL5Z@63TB11SMALLI -
,SMALLINT -
,X'0005 -
,X'0002 -
,X'0000 -
,X'00000025 -
,COL3Z@63TB11IDENT -
,INTEGER -
,X'0003 -
,X'0004 -
,X'0000 -
,N -
,X'00000000 -
,COL4Z@63TB11ROWID -
,ROWID -
,X'0004 -
,X'0011 -
,X'0000 -
,N -
,X'00000000 -
,COL5Z@63TB11SMALLI -
,SMALLINT -
,X'0005 -
,X'0002 -
,X'0000 -
,Y -
,X'00000000 -
,COL6Z@63TB11INT -
,INTEGER -
,X'0006 -
,X'0004 -
,X'0000 -
,Y -
,X'00000000 -
,CHILD, -
  PDDONAA -
  ,Z@63TB21 -
  ,COL3Z@63 -
  ,X'0001 -
  ,COL3Z@63TB21IDENT -
)

```

```

MASKCMD -
  TBCR (PDDONAA) -
  TBNM (Z@63TB11) -
  CLNM (COL2Z@63TB11CHAR2) -
  RULE (MASK,"YA-Za-z0-9"{1,10}", 1, 10) -
  DSN (CKZTRG8.DSNDBC.Z@63DB.Z@63TS1.I0001.A001) -
  ALLP (Y)

```

```

MASKCMD -
  TBCR (PDDONAA) -
  TBNM (Z@63TB11) -
  CLNM (COL6Z@63TB11INT) -
  RULE (RANDOM, 1, 100, YES) -
  DSN (CKZTRG8.DSNDBC.Z@63DB.Z@63TS1.I0001.A001) -
  ALLP (Y)

```

```

MASKCMD -
  TBCR (PDDONAA) -
  TBNM (Z@63TB11) -
  CLNM (COL5Z@63TB11SMALLI) -

```

RULE (SEQUENCE, 1, 1) -
 DSN (CKZTRG8.DSNDBC.Z@63DB.Z@63TS1.I0001.A001) -
 ALLP (Y)

TABLEDEF -
 TBCreator (PDDONAA) -
 TBNAME (Z@63TB21) -
 DBNAME (Z@63DB) -
 TSNAME (Z@63TS2) -
 CCSID (E) -
 OBID (X'0016) -
 REL (Y) -
 COLNO (X'0005) -
 PARENTNO (X'0001) -
 CHILDNO (X'0004) -
 COLUMN (-
 COL1Z@63TB21CHAR -
 ,CHAR -
 ,X'0001 -
 ,X'0028 -
 ,X'0000 -
 ,N -
 ,X'00000025 -
 ,COL2Z@63TB21CLOB -
 ,CLOB -
 ,X'0002 -
 ,X'0004 -
 ,X'0000 -
 ,Y -
 ,X'00000025 -
 ,COL3Z@63TB21IDENT -
 ,INTEGER -
 ,X'0003 -
 ,X'0004 -
 ,X'0000 -
 ,N -
 ,X'00000000 -
 ,COL4Z@63TB21ROWID -
 ,ROWID -
 ,X'0004 -
 ,X'0011 -
 ,X'0000 -
 ,N -
 ,X'00000000 -
 ,COL5Z@63TB21SMALLI -
 ,SMALLINT -
 ,X'0005 -
 ,X'0002 -
 ,X'0000 -
 ,N -
 ,X'00000000 -
 ,PARENT, -
 PDDONAA -
 ,Z@63TB11 -
 ,COL3Z@63 -
 ,X'0001 -
 ,COL3Z@63TB11IDENT -
 ,CHILD, -
 PDDONAA -
 ,Z@63TB41 -
 ,COL5Z@63 -
 ,X'0001 -
 ,COL5Z@63TB41SMALLI -
 ,PDDONAA -
 ,Z@63TB41 -
 ,COL1Z@63 -
 ,X'0002 -

```

,COL1Z@63TB41CHAR -
,COL5Z@63TB41SMALLI -
,PDDONAA -
,Z@63TB51 -
,COL5Z@63 -
,X'0001 -
,COL5Z@63TB51SMALLI -
,PDDONAA -
,Z@63TB52 -
,COL5Z@63 -
,X'0001 -
,COL5Z@63TB52SMALLI -
)

```

```

TABLEDEF -
TBCREATOR (PDDONAA) -
TBNAME (Z@63TB41) -
DBNAME (Z@63DB) -
TSNAME (Z@63TS4) -
CCSID (E) -
OBID (X'004B) -
REL (Y) -
COLNO (X'0005) -
PARENTNO (X'0002) -
CHILDNO (X'0000) -
COLUMN (
COL1Z@63TB41CHAR -
,CHAR -
,X'0001 -
,X'0028 -
,X'0000 -
,Y -
,X'00000025 -
,COL2Z@63TB41CLOB -
,CLOB -
,X'0002 -
,X'0004 -
,X'0000 -
,Y -
,X'00000025 -
,COL3Z@63TB41IDENT -
,INTEGER -
,X'0003 -
,X'0004 -
,X'0000 -
,N -
,X'00000000 -
,COL4Z@63TB41ROWID -
,ROWID -
,X'0004 -
,X'0011 -
,X'0000 -
,N -
,X'00000000 -
,COL5Z@63TB41SMALLI -
,SMALLINT -
,X'0005 -
,X'0002 -
,X'0000 -
,N -
,X'00000000 -
,PARENT, -
PDDONAA -
,Z@63TB21 -
,COL5Z@63 -
,X'0001 -
,COL5Z@63TB21SMALLI -
)

```

```

, PDDONAA -
, Z@63TB21 -
, COL1Z@63 -
, X'0002 -
, COL1Z@63TB21CHAR -
, COL5Z@63TB21SMALLI -
)

```

```

TABLEDEF -
  TBCreator (PDDONAA) -
  TBName (Z@63TB51) -
  DBName (Z@63DB) -
  TSName (Z@63TS5) -
  CCSID (E) -
  OBID (X'005C) -
  REL (Y) -
  COLNO (X'0005) -
  PARENTNO (X'0001) -
  CHILDNO (X'0000) -
  COLUMN (
    COL1Z@63TB51CHAR -
    , CHAR -
    , X'0001 -
    , X'0028 -
    , X'0000 -
    , Y -
    , X'00000025 -
    , COL2Z@63TB51CLOB -
    , CLOB -
  )
  COLNO (X'0005) -
  PARENTNO (X'0001) -
  CHILDNO (X'0000) -
  COLUMN (
    COL1Z@63TB51CHAR -
    , CHAR -
    , X'0001 -
    , X'0028 -
    , X'0000 -
    , Y -
    , X'00000025 -
    , COL2Z@63TB51CLOB -
    , CLOB -
    , X'0002 -
    , X'0004 -
    , X'0000 -
    , Y -
    , X'00000025 -
    , COL3Z@63TB51IDENT -
    , INTEGER -
    , X'0003 -
    , X'0004 -
    , X'0000 -
    , N -
    , X'00000000 -
    , COL4Z@63TB51ROWID -
    , ROWID -
    , X'0004 -
    , X'0011 -
    , X'0000 -
    , N -
    , X'00000000 -
    , COL5Z@63TB51SMALLI -
    , SMALLINT -
    , X'0005 -
    , X'0002 -
    , X'0000 -
    , N -
  )

```

```

,X'00000000 -
,PARENT, -
 PDDONAA -
,Z@63TB21 -
,COL5Z@63 -
,X'0001 -
,COL5Z@63TB21SMALLI -
)

```

```

TABLEDEF -
TBCREATOR (PDDONAA) -
TBNAME (Z@63TB52) -
DBNAME (Z@63DB) -
TSNAME (Z@63TS5) -
CCSID (E) -
OBID (X'0066) -
REL (Y) -
COLNO (X'0005) -
PARENTNO (X'0001) -
CHILDNO (X'0000) -
COLUMN ( -
 COL1Z@63TB52CHAR -
 ,CHAR -
 ,X'0001 -
 ,X'0028 -
 ,X'0000 -
 ,Y -
 ,X'00000025 -
 ,COL2Z@63TB52CHAR16 -
 ,CHAR -
 ,X'0002 -
 ,X'0010 -
 ,X'0000 -
 ,Y -
 ,X'00000025 -
 ,COL3Z@63TB52IDENT -
 ,INTEGER -
 ,X'0003 -
 ,X'0004 -
 ,X'0000 -
 ,N -
 ,X'00000000 -
 ,COL4Z@63TB52ROWID -
 ,ROWID -
 ,X'0004 -
 ,X'0011 -
 ,X'0000 -
 ,N -
 ,X'00000000 -
 ,COL5Z@63TB52SMALLI -
 ,SMALLINT -
 ,X'0005 -
 ,X'0002 -
 ,X'0000 -
 ,N -
 ,X'00000000 -
 ,PARENT, -
 PDDONAA -
,Z@63TB21 -
,COL5Z@63 -
,X'0001 -
,COL5Z@63TB21SMALLI -
)

```

```

/* INDEXSPACES - COPIED SUCCESSFULLY */

```

Chapter 18. Using image copies to clone table spaces and index spaces

You can use DB2 image copies as the source for cloning table spaces and indexes. DB2 Cloning Tool Table Space Cloning reads the appropriate image copies, reads and applies any necessary log records, writes the target data sets, and updates the data pages to make them usable by the target DB2. This feature eliminates the need to access source objects and allows you to clone objects or an entire application at a specific point in time.

Advantages of cloning from image copies include the following:

- You can select various consistent points to clone to (for example, TO_CURRENT, TO QUIESCE, or TO_LOGPOINT), making it easy to clone a set of DB2 objects to a particular point in time.
- When you use image copies as the source of the cloning, the source table spaces and index spaces are not accessed.
- If your site does not have a fast replication tool or you cannot use your fast replication tool for cloning, using image copies as the source can provide a faster cloning process than using traditional “slow” copy procedures.

DB2 Cloning Tool Table Space Cloning's source job for this function differs from the source jobs that do not use image copies as the source. When cloning from image copies, the source job does most of the processing. This includes:

- Locating and reading the last full image copy (both SHRLEVEL REFERENCE and SHRLEVEL CHANGE image copies are supported)
- Applying incremental image copies, if they exist
- Reading and applying log records to the image copy clones to bring the clones to a specific consistent point in time
- Performing updates to make the data usable by the target objects
- Writing the data sets for the target objects

The target job can optionally rebuild indexes and perform other processing that is related to specified cloning options.

Objects that are supported for cloning from image copies are table spaces, index spaces, and LOBs. Indexes can be cloned if image copies are available, or they can be rebuilt.

Restrictions

- Source and target objects must match, or target objects must not exist and must be created by DB2 Cloning Tool Table Space Cloning.
- If a table space was altered to NOT LOGGED between the image copy and the user-specified endpoint, logs can only be applied to the point at which the ALTER was issued.
- Data masking is not supported.
- Cloning directly from a system-level backup (SLB) is not supported. However, you can create image copies from SLBs by using DB2 Recovery Expert, then use those image copies as the source for the clone.
- XML processing is not supported.

- DDL can be generated during cloning from image copies, but the DDL is created from the current catalog definitions.
- The process to clone from image copies cannot use subtasks to multi-task multi-data set non-partitioned objects; therefore, SUBTASKS-DATASET-EXTENSIONS is not allowed.
- The following additional commands are not supported: SIM(A), SUBTASK-DATASET-EXTENSIONS, USE-RUNTIME-REPOSITORY, and any of the DATA-MOVER subcommands such as FASTREP or FCTOPPRCPRIARY.

System requirements

This functionality requires LOG-APPLY processing. During customization of DB2 Cloning Tool Table Space Cloning, you must select the required tasks and steps to create the objects that are needed for LOG-APPLY.

Steps for cloning from image copies (including index rebuilds)

Follow these steps to clone from image copies and rebuild indexes as part of the cloning process.

Before you begin

If you use the ISPF interface to build and generate the cloning jobs, you do not need to follow the steps in this procedure; DB2 Cloning Tool includes the required syntax and DDs as part of the generation process. However, you should ensure that the source subsystem DB2 SDSNEXIT library is in the STEPLIB for the source job.

About this task

These steps assume that:

- You have already set up the basic source and target jobs that contain the control statements to identify the source and target objects and the PARMLIB data sets. Refer to Chapter 15, “Set up procedures for copy by data set for all other methodologies,” on page 199 for information about setting up the jobs.
- You have included the indexes in the LISTDEF and you want to allow DB2 Cloning Tool Table Space Cloning to create and submit index rebuilds. For other ways to rebuild indexes, see “Options for rebuilding indexes when cloning table spaces and index spaces from image copies” on page 262.

Note: If you use the ISPF interface to build and generate the jobs, DB2 Cloning Tool Table Space Cloning adds the required DDs and makes other job changes as part of the job generation process; you do not need to make those changes manually. However, you should carefully review the following steps for other requirements and index rebuild instructions.

Procedure

1. Make the following source job changes:
 - Add DATAMOVER(PGM(SRCIMCPY) and its required parameters to the COPY command.
 - Add LOG-APPLY and the required parameters for cloning from source image copies to the COPY command.
 - Add DDs to the source job as specified in “Source job changes for cloning from image copies” on page 262.

- Ensure that the source subsystem DB2 SDSNEXIT library is in the STEPLIB for the source job.
 - Add or set up the optional parameters to the COPY and/or SET commands for rebuilding indexes. For more information, see “Options for rebuilding indexes when cloning table spaces and index spaces from image copies” on page 262.
2. Submit the following jobs, depending on whether you are cloning within the same or different LPARs:
 - If you are cloning within the same LPAR:
 - a. Submit the source job.
 - b. Submit the target job.
 - If you are cloning across multiple LPARs:
 - a. Submit the target TCP/IP server job (if not currently running). The target TCP/IP server job is required when cloning across LPARs.
 - b. Submit the source job.
 - c. Submit the target job.

Selecting a specific image copy to use as the cloning source

When you specify the end point for the clone, DB2 Cloning Tool Table Space Cloning selects the most recent image copy before that end point as the source for the cloning. However, you might want to select a specific image copy to clone from (and possibly apply logs after that point). Follows these steps to select a specific full image copy as the source for the cloning.

Procedure

1. Query the SYSIBM.SYSCOPY table to find the START_RBA (RBA or LRSN) value (in hexadecimal) of the desired full image copy.
2. In the source job, specify END-POINT (TO_LOGPOINT X'*byte_string*'). For the *byte_string* value, add 1 (in hexadecimal) to the start RBA or LRSN value of the desired image copy.

Note: If you want DB2 Cloning Tool Table Space Cloning to apply log records that were written after the selected image copy, specify a start RBA between the desired image copy and the next image copy.

Results

When the source job is run, DB2 Cloning Tool Table Space Cloning starts with the end point that was provided, scans backwards to locate the image copy, and applies any logs if the end point requires it.

Example

The following sample query could be used to produce a list of image copies and quiesce points for a table space:

```
SELECT
    HEX(START_RBA) AS START_RBA
    , SUBSTR(DSNAME, 1, 44) AS DSNAME
    , ICTYPE
    , DBNAME
    , TSNAME
    , TIMESTAMP
    , SHRLEVEL
```

```

, OTYPE
FROM SYSIBM.SYSCOPY
WHERE ( DBNAME LIKE 'dbname1%'
      AND TSNAME IN ('tsname1'
                    , 'indexname1'
                    , 'tsname2' )
      AND ( SHRLEVEL IN ('R','C') OR ICTYPE = 'Q' ))
ORDER BY OTYPE DESC, TIMESTAMP

```

In the output, the START_RBA column contains the start RBA value in hexadecimal. To select the last full image copy, find the desired image copy in the output and note the start RBA value. Then, add 1 (in hexadecimal) to the start RBA value and include it in the END-POINT parameter.

For example, if the START_RBA of the desired image copy is 00C2E5794C49, specify END-POINT (TO_LOGPOINT X'00C2E5794C4A').

To apply logs, specify an end point between 00C2E5794C49 and the start RBA of the next image copy.

Source job changes for cloning from image copies

Make the following changes to your source job in order to clone from image copies.

- Add the DATAMOVER(PMG(SRCIMCPY) keyword to the COPY command.
- Add the LOG-APPLYsubcommand with the LA-ENABLE(Y) and ZPARM-MEMBER keywords to the COPY command. To specify a consistent point and other parameters, refer to “COPY command syntax” on page 474 for details.
- Add the following DDs to the source job:


```

- //BSDS01 DD DISP=SHR,DSN=h1q.BSDS01
  //BSDS02 DD DISP=SHR,DSN=h1q.BSDS02

```

where *hlq* should be set to the high level qualifier for the source subsystem.

- Add a SYSOUT DD to contain output messages, such as //SYSOUT DD SYSOUT=*
- Add the following SYSINCKZ DD as shown (required for LOG-APPLY):


```

SYSINCKZ DD DISP=SHR,DSN=h1q.LOGAPCTL(membername)

```
- Ensure that the DB2 SDSNEXIT library is in the STEPLIB for the source job.

Options for rebuilding indexes when cloning table spaces and index spaces from image copies

When using image copies as the cloning source, indexes can be cloned if the image copies are available, or they can be rebuilt. Several alternative options for creating and submitting the JCL to rebuild indexes are described in this topic.

Note:

If the indexes are not included in the LISTDEF, they are not refreshed and therefore must be rebuilt using job templates; refer to the steps in Method 1.

Method 1: Using job templates and the JOB-TEMPLATE command

For more information about job templates, refer to Chapter 20, “Using job templates,” on page 269.

1. Add JOB-TEMPLATE (*inddname1*, *outddname1*) to the COPY command.
2. Modify the sample templates. Make a copy of either the CKZJOB1 (recommended) or CKZJOB2 template in the SCKZJCL library and modify it for your site. The template contains the statements used to build the REBUILD INDEX utility job. Instructions for updating the template are contained in the member. Two template pairs can be specified, one using REBUILD INDEX (ALL) TABLESPACE and another using REBUILD INDEXSPACE.
3. When you submit the source job, the rebuild JCL is created and placed in *outddname1* as specified in the JOB-TEMPLATE command.
4. To rebuild the indexes, submit the JCL in *outddname1*.

Method 2: Using TARGET-JOB-INDEX-REBUILD-DDN

This method can rebuild indexes defined as COPY NO and can be submitted as part of the target job, if desired.

1. Add TARGET-JOB-INDEX-REBUILD-DDN to the source job.
2. Modify the sample template:
 - a. Make a copy of the CKZJOB1 member (required for PGM(SRCIMCPY)) in the SCKZJCL library and modify it for your site. This template contains the statements used to build the REBUILD INDEX utility job. Instructions for updating the template are contained in the member.
 - b. Add the following two DDs to the target job for rebuilding indexes:

```
//ddnameI DD DISP=SHR,DSN=hlq.indsn(mbr)
```

where *ddname* is the DD specified in the TARGET-JOB-INDEX-REBUILD keyword. You must append an I to *ddname*. *hlq.indsn(mbr)* is the data set location and member name of the template you edited.

```
//ddnameO DD DISP=OLD,DSN=hlq.outdsn(mbr)
```

where *ddname* is the DD specified in the TARGET-JOB-INDEX-REBUILD keyword. You must append an O to *ddname*. *hlq.outdsn(mbr)* is the data set location and member name where you want the utility job to be saved. This DD will contain the output from the generated REBUILD INDEX job.
3. (Optional) If you want the index rebuilds to be submitted as part of target job, specify SET command REBUILD-INDEXES-EXECUTE(Y) in the source job.
4. (Optional) If you want indexes defined as COPY NO to be rebuilt, specify the LOG-APPLY command REBUILD-COPY-NO-INDEXES(Y) in the source job.
5. When the target job is executed, the job to rebuild indexes is generated and placed in the *ddnameO* output data set specified in the target job. If you did not specify the job to be submitted as part of the target job, submit the job to rebuild indexes.

Chapter 19. Using LOG-APPLY to make consistent copies of table spaces and index spaces

The LOG-APPLY keyword allows the DB2 log records written to the source objects to be applied to the target objects during target job processing.

Any updates that are made to the source objects in the time between submitting the source job and running the target job are applied to the target objects. LOG-APPLY provides you with a way to consistently clone a set of related DB2 objects from source table spaces to target table spaces without stopping them or making them unavailable for updates. The LOG-APPLY procedure varies depending on whether you are cloning between or within subsystems on the same LPAR of a z/OS sysplex, or if you are cloning between DB2 subsystems on different LPARs or on different z/OS sysplexes.

- When cloning across multiple LPARs, LOG-APPLY runs on the target job after the source job has completed. A target TCP/IP server job is required for the source job, and a source TCP/IP server job is required for the target job. The target TCP/IP server job connects to the target DB2 subsystem; the source TCP/IP server job connects to the source DB2 subsystem.
- When cloning within the same or different LPAR, you can use LOG-APPLY to copy image copy data sets from the source subsystem to the target subsystem. The target TCP/IP server must be specified if copying across LPARs.
- When cloning within the same LPAR. The LOG-APPLY process runs in the target job, but does not need a Source TCP/IP server job.

You can minimize the amount of time it takes to process DB2 logs by specifying a QUIESCE as part of LOG-APPLY and by running the target job in a timely manner.

Restriction: Logs cannot be applied for LOBs, XML spaces, and index spaces.

System requirements

During customization of DB2 Cloning Tool, you must select the required tasks and steps to create the objects needed for LOG-APPLY.

Before you begin

If you use the ISPF interface to build and generate the cloning jobs, you do not need to follow the steps in this procedure; DB2 Cloning Tool includes the required syntax and DDs as part of the generation process. However, you should carefully review the following steps for other requirements and instructions for rebuilding indexes.

Summary of steps

1. Add LOG-APPLY and other keywords and their required parameters to the COPY and SET commands.
2. Add DDs to the source job. Add or set up the optional parameters to the COPY and/or SET commands for rebuilding indexes. For more information, see “Options for rebuilding indexes” on page 267.
3. Add DDs to the target job.

4. Submit the following jobs, depending on whether you are cloning within the same or different LPARs:
 - If you are cloning within the same LPAR:
 - a. Submit the source job.
 - b. Submit the target job.
 - If you are cloning across multiple LPARs:
 - a. Submit the source and target TCP/IP server jobs (if not currently running). The source TCP/IP server job is required when cloning across LPARs using LOG APPLY.
 - b. Submit the source job.
 - c. Submit the target job.

JCL library members CKZ5SRC and CKZ5TRG contain examples of setting up a job using log apply.

Source job changes

Make the following changes to your source job in order to apply DB2 logs.

- Add the LOG-APPLY subcommand with the LA-ENABLE(Y) and ZPARAM-MEMBER keywords' and their parameters to the COPY command; refer to "COPY command syntax" on page 474 for information about the keywords and parameters. If you are cloning across multiple LPARs, you must specify the USE-TCPIP (Y) parameter in the LOG-APPLY keyword.
- Add the following DDs to the source job:

```
//BDS01 DD DISP=SHR,DSN=hlq.BSDS01
//BDS02 DD DISP=SHR,DSN=hlq.BSDS02
```

where *hlq* should be set to the high level qualifier for the source subsystem.
- Ensure that the source subsystem DB2 SDSNEXIT library is in the STEPLIB for the source job.

Source TCP/IP server job (required when using LOG-APPLY across multiple LPARs)

CKZTCPSS, the source TCP/IP server job, is required for LOG-APPLY when the source and target DB2 systems are on different LPARs.

Configure the JCL for the source TCP/IP server job and submit it to run on the same LPAR where the source DB2 subsystem is running. The DB2 Cloning Tool target job connects to the source TCP/IP server job that is running on the same LPAR as the source DB2, and then instructs the server to do the following:

- Connect to the source DB2 subsystem
- Determine the logs records that need to be extracted
- Read those log records, and process them for delivery to the target job
- Send the log records back to the target job via TCP/IP

The source TCP/IP server job must be started before submitting the target job that will make the spaces accessible on the target DB2 subsystem. When submitted, the target job applies the log records to the copied data sets, bringing the cloned objects to a consistent point in time.

SET commands for the source TCP/IP server job include:

- SOURCE-TCPIP-SERVER-JOB – specify the source TCP/IP server job.
- SOURCE-TCPIP-SERVER-PORT – specify the connection port in the source TCP/IP server.
- SOURCE-TCPIP-STC-NAME – specify the TCP/IP address space name in the target job.
- SOURCE-CONNECT-DB2-ON-CLIENT-CONNECT – specify whether or not to disconnect from DB2 on TCP/IP disconnect.
- SOURCE-IP-VERSION6 – Specify N to indicate IP v4.

See the command reference for details about these commands.

Target job changes

Make the following changes to your target job in order to apply DB2 logs.

- Ensure that the source subsystem DB2 SDSNEXIT library is in the STEPLIB for the target job.
- Add the following DDs to the target job. The data set names are required to be as shown.

```
//SYSINCKZ DD DISP=SHR,DSN=hlq.LOGAPCTL(membername)
//SYSOUT DD SYSOUT=*
//INFOM DD SYSOUT=*
```

- Add the following two DDs to the target job for rebuilding indexes:

```
//ddnameI DD DISP=SHR,DSN=hlq.indsn(mbr)
```

where *ddname* is the DD specified in the TARGET-JOB-INDEX-REBUILD keyword. You must append an I to *ddname*. *hlq.indsn(mbr)* is the data set location and member name of the template you edited.

```
//ddnameO DD DISP=OLD,DSN=hlq.outdsn(mbr)
```

where *ddname* is the DD specified in the TARGET-JOB-INDEX-REBUILD keyword. You must append an O to *ddname*. *hlq.outdsn(mbr)* is the data set location and member name where you want the utility job to be saved. This DD will contain the output from the generated REBUILD INDEX job.

- If you are cloning across multiple LPARs, parameters used by the target job for TCP/IP communications to the source TCP/IP server job are passed to the target job from the source job using the SYNCDB2 data set. Include these parameters for the SET command:
 - SOURCE-TCPIP-SERVER-PORT - Connection port in the source TCP/IP server
 - SOURCE-TCPIP-STC-NAME - TCP/IP address space name in the target job
 - SOURCE-TCPIP-SERVER-IP – IP of the source TCP/IP server job

Options for rebuilding indexes

Indexes can be cloned or they can be rebuilt. Several alternative options for creating and submitting the JCL to rebuild indexes are described in this topic.

Note: If the indexes are not included in the LISTDEF or included using the copy command keyword ALWAYS-COPY-INDEXSPACES(Y), they are not refreshed and therefore must be rebuilt using job templates; refer to the steps in Method 1.

Method 1: Using job templates and the JOB-TEMPLATE command to rebuild indexes

Indexes can be cloned or they can be rebuilt. This method uses job templates and the JOB-TEMPLATE command to rebuild indexes.

Procedure

1. Add JOB-TEMPLATE (*inddname1*, *outddname1*) to the COPY command.
2. Modify the sample templates. Make a copy of either the CKZJOB1 (recommended) or CKZJOB2 template in the SCKZJCL library and modify it for your site. The template contains the statements used to build the REBUILD INDEX utility job. Instructions for updating the template are contained in the member. Two template pairs can be specified, one using REBUILD INDEX (ALL) TABLESPACE and another using REBUILD INDEXSPACE.
3. When you submit the source job, the rebuild JCL is created and placed in *outddname1* as specified in the JOB-TEMPLATE command.
4. To rebuild the indexes, submit the JCL in *outddname1*.

Method 2: Using TARGET-JOB-INDEX-REBUILD-DDN to rebuild indexes

This method can rebuild indexes defined as COPY NO and can be submitted as part of the target job, if desired.

Procedure

1. Add TARGET-JOB-INDEX-REBUILD-DDN to the source job.
2. Modify the sample template:
 - a. Make a copy of the CKZJOB1 member (required for PGM(SRCIMCPY)) in the SCKZJCL library and modify it for your site. This template contains the statements used to build the REBUILD INDEX utility job. Instructions for updating the template are contained in the member.
 - b. Add the following two DDs to the target job for rebuilding indexes:

```
//ddnameI DD DISP=SHR,DSN=hlq.indsn(mbr)
```

where *ddname* is the DD specified in the TARGET-JOB-INDEX-REBUILD keyword. You must append an I to *ddname*. *hlq.indsn(mbr)* is the data set location and member name of the template you edited.

```
//ddnameO DD DISP=OLD,DSN=hlq.outdsn(mbr)
```

 where *ddname* is the DD specified in the TARGET-JOB-INDEX-REBUILD keyword. You must append an O to *ddname*. *hlq.outdsn(mbr)* is the data set location and member name where you want the utility job to be saved. This DD will contain the output from the generated REBUILD INDEX job.

3. Optional: If you want the index rebuilds to be submitted as part of target job, specify SET command REBUILD-INDEXES-EXECUTE(Y) in the source job.
4. Optional: If you want indexes defined as COPY NO to be rebuilt, specify the LOG-APPLY command REBUILD-COPY-NO-INDEXES(Y) in the source job.
5. When the target job is executed, the job to rebuild indexes is generated and placed in the *ddnameO* output data set specified in the target job. If you did not specify the job to be submitted as part of the target job, submit the job to rebuild indexes.

Chapter 20. Using job templates

Job templates consist of the z/OS JCL statements, DSS commands, user and processing variables that DB2 Cloning Tool Table Space Cloning uses for input. The job statements generated are then written to the output DD specified in the JOB-TEMPLATE subcommand.

Refer to the sample jobs CKZJOB1, CKZJOB2, and CKZJOB3, provided in the JCL library. They contain detailed information about setting up a job template.

| During the table space cloning process, job templates also can be used to generate
| utility jobs for target objects. Utility job templates are provided for CHECK DATA,
| CHECK INDEX, REBUILD INDEX, QUIESCE, REORG INDEXSPACE, REORG
| TABLESPACE, and RUNSTATS utilities.

The EMC API is supported using job templates. It is more restrictive in how the variables are used. Use sample job CKZJOB3 to set up the EMC API template. In addition, this job can only be submitted after the source job has run with DATA-MOVER (PGM(NONE)).

For additional information about specific commands and parameters used for job templates, refer to the following:

JOB-TEMPLATE

This parameter provides the DD names that contain the templates. See the "COPY" on page 473 command topic for more information.

CMDDNAME (DATA-MOVER subcommand)

This parameter provides the name of an output data set that contains a job built using the JOB-TEMPLATE. See the "COPY" on page 473 command topic for more information.

TEMPLATE-VARIABLE

This parameter allows variables to be set for the life of the source job. See the "SET" on page 516 command topic for more information.

Predefined processing variables

Predefined processing variables control how DB2 Cloning Tool Table Space Cloning generates the JCL. They are used to build DSS input command stream and job steps.

&&BEGDSSCC

Begin adding DSS continuation characters for data set list variables &SRCDSNL, &TRGDSNL and &PAIRDSNL.

&&BEGDSSCM

(Required) Begin DSS commands. This should be the first variable after the SYSIN DD.

&&BEGDSSRC

(Required) Begin DSS repeat commands. This is the next variable after DSS commands that only execute one time per step. The DSS COPY command would be included after this variable.

&&BEGIDSPB

Start an individual data set processing block. Required for individual data set processing variables if there are cards without variables with the data set variables. For example, there might be an open parenthesis on a line by itself.

&&BEGPPAR

Add a parenthesis before the source and after the target data set when processing source target pairs. Required when using DSS RENAMEU, for example. Applies to data set list variables only, not individual data set processing variables.

&&BEGSTEP

(Required) Begin step generation. This must be before the z/OS EXEC JCL statement.

&&ENDDSSCC

End adding DSS continuation characters.

&&ENDDSSCM

(Required) End DSS commands.

&&ENDDSSRC

(Required) End DSS repeat commands.

&&ENDIDSPB

End an individual data set processing block.

&&ENDPPAR

Stop adding a parenthesis when processing source target pairs.

&&ENDSTEP

(Required) End step generation.

&&HEADER

Place this on a line at the top of the template on a line beginning with `/**`. It adds the date, time, job name and job number of the source job on that line. For example:

```
/** 08292 20:59:03.53  JOBNAME=PDDONAX8  JOBID=J0202482
```

&&PAIRDSNL

Data set pair (source and target) list. The character following this variable is the suffix character and will be added after each data set except the last target data set. No other characters can be on the line.

&&SRCDSN0 and &&SRCDSN1

Source data set. Each occurrence will output the next source data set. Use this variable along with `&&TRGDSN0/1` to code the exact number and position of each data set in the DSS command string. The maximum number in one template is 255 as that is the maximum number of data sets DSS can process in a single COPY command. These variables provide more flexibility. They can, for example, be used to copy the target data sets back to the source data sets. `&&PAIRDSNL` only copies from the source to the target.

&&SRCDSNL

Source data set list. Each source data set will be output in the same order as the internal DB2 Cloning Tool Table Space Cloning copy command. The character following this variable is the suffix character and will be added after each data set except the last. No other characters can be on the line.

| **&&SRCOBJS**
| Source objects.

| **&&TRGOBJS**
| Target objects.

| **&&SRCSSID**
| Source DB2 SSID.

| **&&STEPNUM**
| Add this on the step execution card to request a three-digit number
| appended to the step name. For example:
| //STEP.&&STEPNUM EXEC PGM=ADRDSSU,REGION=4M

Note: The dot between STEP and &&STEPNUM indicates the variable
 value is to be concatenated with the preceding string.

| **&&TRGDSN0 and &&TRGDSN1**
| Target data set. Each occurrence will output the next target DSN. Use this
| variable along with &&SRCDSN0/1 to code the exact number and position
| of each data set in the DSS command string. The maximum number in one
| template is 255 as that is the maximum number of data sets DSS can
| process in a single COPY command.

| **&&TRGDSNL**
| Target data set list. Each target data set will be output in the same order as
| the internal DB2 Cloning Tool Table Space Cloning copy command. The
| character following this variable is the suffix char and will be added after
| each DSN except the last. No other characters can be on the line.

| **&&TRGSSID**
| Target DB2 SSID.

| **&&TRGVCAT**
| Target VCAT (DEFVCAT in the TARGET-DB2 command).

Data set processing variables

There are seven data set processing variables. Each must begin on a separate line and may be followed by one character, a comma or a close parenthesis used to terminate the DSS data set subcommand.

| **&&SRCDSNL, &&TRGDSNL, &&PAIRDSNL**
| These variables use the PARMLIB member value of DSNS_PER_COPY to
| determine how many data sets are included with each DSS COPY
| command.

| **&&SRCDSN0/1, &&TRGDSN0/1**
| These variables determine the number of data sets per COPY command by
| the number of variables found. For example, if the parameter value for the
| number of data sets per DSS COPY command is 255 and 50 &&SRCDSN
| variables are found, 255 is reduced to 50 while this template is being
| processed. It is changed back to 255 for the next template (if any). If more
| than 255 &&SRCDSN0/1 or &&TRGDSN0/1 variables are found, the job
| ends with an error.

Block processing variables

There are three sets of required block processing variables: z/OS JCL step, DSS commands and DSS repeat commands.

The block processing variables must be present in every template and must be in the correct order. They are in order of appearance in the template:

- &&BEGSTEP
- &&BEGDSSCM
- &&BEGDSSRC
- &&ENDDSSRC
- &&ENDDSSCM
- &&ENDSTEP

Other block processing variables control DSS continuation characters and when to enclose data sets in parentheses. They are:

- &&BEGDSSCC
- &&BEGIDSPB
- &&BEGPPAR
- &&ENDDSSCC
- &&ENDIDSPB
- &&ENDPPAR

Job templates for utilities

During the table space cloning process, job templates can be used to generate utility jobs for target objects. Utility job templates are provided for CHECK DATA, CHECK INDEX, REBUILD INDEX, QUIESCE, REORG INDEXSPACE, REORG TABLESPACE, REPAIR, and RUNSTATS utilities.

The utility job templates are located in the SCKZJCL library. The following table describes the utility jobs and their member names.

Table 44. Utility job templates and associated member names

DB2 utility	Member name
CHECK DATA	CKZJOB CD
CHECK INDEX	CKZJOB CI
QUIESCE	CKZJOB Q
REBUILD INDEX	CKZJOB I, CKZJOB R
REORG INDEXSPACE	CKZJOB RI
REORG TABLESPACE	CKZJOB RO
REORG TABLESPACE (DB2 V11 only, when converting from 6-byte to 10-byte RBA/LRSN)	CKZJOB RX
REPAIR	CKZJOB RP
RUNSTATS	CKZJOB RS

Use the templates as follows:

1. Edit the utility template by following the instructions in the member. Generally, you will add a job card, modify the job's user ID and data set high level qualifiers, and add DDs to the source job that will contain the generated utility JCL.
2. Submit the source job.

3. Submit the target job.
4. Verify that the target objects have been successfully created on the target subsystem.
5. Submit the utility job(s).

Variables in utility job templates

The variables used in the utility job templates are as follows:

&&JOB CARD

This variable is used to indicate the end of job card information.

&&DATABASE

This variable is used to insert a database name in the utility command.

&&TABLESP

This variable is used to insert a table space name in the utility command.

&&OBJECTTY

This variable is used to insert TABLESPACE or INDEXSPACE in the utility command. This variable allow you to use the same template for both table spaces and index spaces. Refer to the example that follows for more information.

&&OBJECTSP

This variable is used to insert a table space or index space name in the utility command. This variable allow you to use the same template for both table spaces and index spaces. Refer to the example that follows for more information.

&&INDEXSP

This variable is used to insert an indexspace name in the utility command.

&&TRGSSID

This variable defines the target subsystem on which the utility will be run.

&&TRGOBJS

This variable defines the target objects on which the utility will be run.

&&INCRVAL

This variable is used as an incremental in constructing data set names used during the utility, such as SYSREC and SYSUT1.

Using the &&OBJECTTY and &&OBJECTSP variables allows both table spaces and index spaces to be processed in the same job template. For example, in a REPAIR utility template, you can either:

- Create two separate templates, one for table spaces and one for index spaces, with REPAIR syntax as follows:

```
REPAIR CATALOG TABLESPACE  
                &&DATABASE.&&TABLESP
```

```
REPAIR CATALOG INDEXSPACE  
                &&DATABASE.&&INDEXSP
```

- Or, you can create one template with the following REPAIR syntax:

```
REPAIR CATALOG &&OBJECTTY  
                &&DATABASE.&&OBJECTSP
```

Chapter 21. Using the ISPF interface

DB2 Cloning Tool offers an ISPF interface that allows you to create subsystem and table space cloning jobs using interactive panels. This topic describes how to use the ISPF interface for DB2 Cloning Tool.

The DB2 Cloning Tool ISPF interface

You can use the DB2 Cloning Tool ISPF interface to create the JCL and control cards required to clone DB2 subsystems and to clone table spaces and index spaces. The menu-driven interface allows you to easily create cloning jobs with specific command parameters, and then save that information in profiles that can be used again. In addition, subsystem information can be configured once and then is available to all users of the interface.

Starting the interface

Note: Before attempting to use the ISPF interface, ensure that customization steps been completed as described in Chapter 4, “Customizing DB2 Cloning Tool,” on page 51 and Chapter 5, “After customizing DB2 Cloning Tool using Tools Customizer,” on page 71.

Start the interface using the provided CLIST by using the command TS0 CKZ on the command line.

Note: The DB2 Cloning Tool ISPF interface requires a minimum region size of 30000 KB.

The DB2 Cloning Tool Primary Option Menu

The DB2 Cloning Tool Primary Option Menu is the starting point for all cloning functions. The Primary Option Menu, shown in the following figure, is displayed when you start the DB2 Cloning Tool ISPF interface.

```
DB2 Cloning Tool for z/OS      Primary Option Menu
Option ===>

0  User settings                User ID . . . : TWUSR
1  Clone                       System ID . . . : RS22
2  Administrator functions      Appl ID . . . : CKZ

X  Exit                         Version . . . : 3.1
```

From the Primary Option Menu, you can perform the following actions by entering the corresponding option number in the **Option** field and pressing Enter:

0 - User settings

Select this option to specify defaults for command parameters, work data sets, other settings for both subsystem cloning and table space cloning. These defaults are used for creating cloning profiles.

1 – Clone

Select option 1 to create cloning profiles for subsystem or table space cloning.

2 – Administrator functions

Select option 2 to add or configure DB2 subsystems to be used as source and targets for the cloning process. DB2 subsystems should be configured before attempting to create cloning profiles.

X – Exit

Select this option to exit the ISPF interface.

About cloning profiles

All the settings required to build the jobs for subsystem or table space cloning are saved in VSAM profiles. You can create profiles that can be shared with other users, or you can specify that profiles be read-only or completely inaccessible by others.

About the ISPF help system

ISPF help panels are available. Every product panel in DB2 Cloning Tool has its own help panel. The help panel lists the purpose of the panel, available commands, and fields and column data that is displayed. Enter HELP or press PF1 to display a help panel.

Detailed information about command parameters and other valid values on the panels are listed in the help panels. Except for configuring user settings, the panel values are not described in detail in these topics. Use the help system or the command reference topics that follow if you need more information.

- Chapter 23, “DB2 Cloning Tool Subsystem Cloning commands,” on page 337
- Chapter 24, “DB2 Cloning Tool Table Space Cloning commands,” on page 473

Configuring DB2 subsystems

The Administrator functions option on the Primary Option Menu allows you to configure all DB2 subsystems that might be used by subsystem or table space cloning procedures.

You must define the DB2 subsystem information for your site on these panels before attempting to create subsystem or table space cloning jobs.

Enter 2 on the DB2 Cloning Tool Primary Option Menu menu to access administrator functions. The Administrator functions panel is displayed, as show in the following figure:



Configuring a subsystem

To create a new subsystem or configure a subsystem that you have already created, follow these steps.

Procedure

1. On the Administrator functions menu, enter option 1.
2. On the DB2 subsystems panel, enter C in the Command field. The Enter New DB2 Subsystem Profile Options window is displayed.
3. On the Enter New DB2 Subsystem Profile Options window, enter the DB2 subsystem ID.
4. Press Enter. The Edit DB2 Subsystem panel is displayed.
5. Specify DB2 load library names for the subsystem being defined.
6. If the subsystem will be used for subsystem cloning, enter 1 in the Option line to specify information required for cloning DB2 subsystems and press Enter.
7. If the subsystem will be used for table space cloning, enter 2 in the Option line to specify information required for table space cloning and press Enter.

Specifying information for subsystem cloning

To successfully generate JCL that uses this subsystem as a target for subsystem cloning jobs, you must provide the system VCAT and the special ZPARMs member.

Refer to the product help panel for more detailed information about the fields on this panel.

Enter 1 on the Edit DB2 Subsystem menu. The Subsystem cloning information panel is displayed, as shown in the following figure:

```
Subsystem cloning information
Command ==>
SSID . . . . . : SS01
Description . . :
Use as Subsystem Cloning Source or Target only . .      More:      +
                                                         (SOURCE, TARGET,
                                                         or blank)
System ID where this DB2 normally runs . .
Group name . . . . . (if data sharing)
Member name . . . . . (if data sharing)
Special ZPARMs member . . . . .
System VCAT . . . . . MVFQTRG8

DDF:
LOCATION . . . . . RS23D81A
GENERIC . . . . .
LUNAME . . . . .
PASSWORD . . . . .
PORT . . . . . 6002 (1-65535, or blank)
RESPORT . . . . . (1-65535, or blank)
SECPORT . . . . . (1-65535, or blank)
IPNAME . . . . .
IPV4 . . . . . 192.168.55.23
IPV6 . . . . .
GRPIP4 . . . . .
GRPIP6 . . . . .
ALIAS1 . . . . .
ALIAS2 . . . . .
ALIAS3 . . . . .
ALIAS4 . . . . .
ALIAS5 . . . . .
```

Specifying information for table space cloning

This topic describes how to specify subsystem information for DB2 table space cloning in the administrative options.

Enter 2 on the Edit DB2 Subsystem menu. The Tablespace cloning information panel is displayed:

```

                                Tablespace cloning information
Option ==>

SSID . . . . . : SS01
Description . . :

Default VCAT . . FTRQTRG8

DDF:
LOCATION . . . . RS23D81A
USERID . . . . .
PASSWORD . . . .

TCPIP Server:
PORT . . . . . 6002 (1-65535, or blank)
IPV4 . . . . . 192.168.55.23
IPV6 . . . . .
```

Refer to the product help panel for more detailed information about the fields on this panel.

Configuring user settings

The User Settings option on the Primary Option Menu allows you to specify defaults that are used when creating profiles. Defaults can be set for commands, work data sets, and job cards for both subsystem cloning and table space cloning.

Defaults are originally derived from the CKZINI PARMLIB member, but can be customized for each user ID using these panels. The defaults are saved in ISPF profiles and are specific to each TSO user ID. This allows different users to have different defaults when creating cloning profiles.

Enter 0 on the DB2 Cloning Tool Primary Option Menu to access user options. The User Settings panel is displayed, as shown in the following figure:

```

DB2 Cloning Tool for z/OS      User Settings
Option ==>

0 User Options
1 User DB2 subsystem clone settings
2 User DB2 tablespace clone settings
```

Setting user defaults (job card and work data sets)

Follow these steps to set general defaults settings that apply to both volume and table space cloning. These include a default job card and default work volumes.

Procedure

1. Enter 0 on the User Settings menu to access user options. The Set Processing Options menu is displayed.
2. On the Set Processing Options panel, enter 1 to specify job card options and press Enter. The Set Batch Job Card Information panel is displayed.

3. Enter job card information for your site.
4. Press **PF3 (END)** to return to the Set Processing Options panel.
5. On the Set Processing Options panel, enter **2** in the **Option** field and press Enter. The User Options panel is displayed.
6. Enter default values for the work unit and work volume unit. Press **PF3 (END)** when you are finished.

Setting subsystem cloning defaults

This topic describes how to specify defaults for subsystem cloning commands and work data sets.

Defaults are originally derived from the CKZINI PARMLIB member, but can be customized for each user ID using these panels. These defaults are then saved in ISPF profiles and are specific to each TSO user ID. This allows different users to have different defaults when creating cloning profiles.

Enter **1** on the User Settings menu to access subsystem cloning defaults. The User DB2 subsystem clone settings panel is displayed, as shown in the following figure:

```

User DB2 subsystem clone settings
Option ==>

Subsystem Clone Profile Default Values:
  Prefix for work data sets . .
  Work data sets unit device . .      (SYSDA, DISK, or etc.)

Valid command selection values are
  1 COPY command
  2 COPYCHECK command
  3 RENAME command
  4 DB2FIX command
  5 DB2SQL command
  6 DB2START command
  7 DB2STOP command
  8 DB2UPDATE command
  9 DB2UTILXCLEAN command
 10 RESTORE-FROM-DUMPTAPES command

```

Subsystem clone profile default values

Prefix for work data sets

Enter the prefix you would like to use for work data sets that might be needed when cloning a DB2 subsystem.

Work data sets unit device

Enter a valid unit device that will hold the work data sets.

Command defaults

To set default values for each command, enter the appropriate number for the command in the **Option** field.

COPY command defaults

COPY invokes volume copies via FlashCopy or SnapShot if the DATA-MOVER(PGM(ADRDSSU)) is specified, or invokes volume copies via TimeFinder/Clone mainframe SNAP Facility if the DATA-MOVER(PGM(EMCSNAP)) is specified.

The following values can be set on the subsystem cloning COPY command defaults panel. These values are used strictly as defaults for COPY command parameters when you are creating a new subsystem cloning profile. The parameter values can later be edited in the profile.

TARGET-VOLS-SHOULD-BE-EMPTY

Specify whether a check should be performed during the volume pairing process to ensure the target volumes are empty before issuing FlashCopy or SnapShot. In the event a subsequent RENAME fails and the COPY must be rerun, the target volumes will not be cleaned off if YES was specified for this parameter. Either initialize the target volumes or change this field to NO.

CATWORK-DSN Mask

Specify a mask to be used to derive data set names for catalog backup data sets dynamically allocated during the COPY step. The mask must include an asterisk (*) as one qualifier. Data sets will be created by substituting two eight-byte qualifiers in place of the provided asterisk. Hence, because 17 bytes (8 the dot 8) of the name will be generated, you are responsible for the resolved names not exceeding 44 characters.

For example, a CATWORK-DSN mask of SITENAME.ABC.CATWORK.* will cause data sets to be created such as:
SITENAME.ABC.CATWORK.UCATBKUP.BKP00001

The asterisk in the mask does not need to be the lowest level qualifier. For example, a CATWORK-DSN mask of SITENAME.ABC.CWORK.*.DATA is valid.

CATWORK-ATTR

Specify the allocation attributes used when catalog backup data sets are dynamically allocated. Allocation attributes are specified in TSO allocate syntax (e.g., UNIT(SYSDA) SPACE(1 1) TRACKS, etc.). The attributes that can be specified are:

- DATACLAS(data class name)
- MGMTCLAS(management class name)
- SPACE(quantity increment)
- STORCLAS(storage class name)
- TRACKS/CYLINDERS
- UNIT(unit)
- VOLUME(serial)

If an initial attempt running COPY fails because a catalog backup data set exceeds extents, increase the allocation and run again. Once successful, examine the space actually used and decrease if desired. To accommodate a future increase in the size of catalogs, leave the allocation with room to spare.

COPYCHECK command defaults

COPYCHECK provides a mechanism to either wait for copies to complete, or to terminate previously established volume relationships.

The following values can be set on the subsystem cloning COPYCHECK command defaults panel. These values are used strictly as defaults for COPYCHECK command parameters when you are creating a new subsystem cloning profile. The parameter values can later be edited in the profile.

WAIT time

Specify the maximum time in minutes that COPYCHECK should continue checking at 30-second intervals to see if copy relationships have completed for all volume copies initiated in a corresponding COPY step.

WAIT RC

Specify the return code to be used if the specified time limit expires before all copies are complete and COPYCHECK terminates.

RENAME command defaults

The RENAME command renames and catalogs the data sets from the COPY command onto target volumes.

The following values can be set on the subsystem cloning RENAME command defaults panel. These values are used strictly as defaults for RENAME command parameters when you are creating a new subsystem cloning profile. The parameter values can later be edited in the profile.

EXCLUDE-SRCNAME

This default setting specifies the return code that will be given for data sets that match entries in the EXCLUDE-SRCNAME-MASKS keyword. 0 specifies that a return code of zero will be given for data sets that match entries in the EXCLUDE-SRCNAME-MASKS keyword. NOTRENAMED-RC specifies that the RC specified in the NOTRENAMED keyword will be given for data sets that match entries in the EXCLUDE-SRCNAME-MASKS keyword. Using RC(0) addresses the situation where there are known data sets on the volumes that will not be renamed and it is desired to use NOTRENAMED(RC(8)) to know if some not known data sets are on the volumes.

GDG-ALL-MIGRATED**GDG-ALL-MIGRATED RETAIN RC**

These default settings support DFSMSHsm, FDR, and CADisk. GDG-ALL-MIGRATED addresses the situation where a GDG matches a RENAME mask and all the source generations have been migrated. Specify SKIP to skip the migrated GDG entry. Specify RETAIN to keep the GDS entries in the GDG base record. If you enter RETAIN, enter a corresponding return code of 0 or 4 in the GDG-ALL-MIGRATED RETAIN RC field.

Note: If RETAIN is used, because the migrated generations do not exist under the new name, subsequent access to the generations will fail whether one is accessed specifically or via specification of the base name only. This option is provided to retain relativity.

GDG-EMPTY**GDG-EMPTY RETAIN RC**

These default settings address an empty base GDG that matches a RENAME mask. In the GDG-EMPTY field, enter SKIP to skip GDG entry, or enter RETAIN to add the new base entry to the target user catalog. If you specify RETAIN, enter a corresponding return code of 0 or 4 in the GDG-EMPTY RETAIN RC field.

GDG-MIGRATED**GDG-MIGRATED RETAIN RC**

These default settings address the situation where a GDG matches a rename mask and at least one generation is indeed found on a volume, yet one or more generations are migrated. If you want to treat the migrated generation as an error, enter ERROR in the GDG-MIGRATED field. To keep

the GDS entry in the GDG base record, enter RETAIN. If you enter RETAIN, enter a corresponding return code of 0 or 4 in the GDG-MIGRATED RETAIN RC field.

Note: If RETAIN is used, because the migrated generation does not exist under the new name, subsequent access to the generation will fail whether it is accessed specifically or via specification of the base name only.

GDG-TAPE

GDG-TAPE RETAIN RC

These settings address the situation where a GDG matches a rename mask and at least one generation is indeed found on a volume, yet one or more generations are on tape. To treat the tape generation as an error, enter ERROR in the GDG-TAPE field. To keep the GDS entry in the GDG base record, enter RETAIN. If you enter RETAIN, enter a corresponding return code of 0 or 4 in the GDG-TAPE RETAIN RC field.

Note: If RETAIN is specified, accessing a target tape GDS will cause a S813 ABEND whether it is accessed specifically or via specification of the base name only. To avoid destroying the relativity of active generations, removing selected generations is not allowed. Retaining non-existent tape generations may be suitable for situations such as overstated GDG limits where older generations may have been created on tape.

ISSUE-VCLOSE

Specify whether a catalog modify command, F CATALOG,VCLOSE(targetvolser), will be issued as part of the volume RENAME processing. The Catalog Address Space (CAS) caches VVDS information. The modify command requests that the VVDS information cached for the target volume be refreshed. NO specifies that the modify command will NOT be issued. BEFORE specifies that the modify command will be issued only before the VVDS is updated. AFTER specifies that the modify command will be issued only after the VVDS has been updated. YES specifies that the modify command will be issued both before the VVDS is updated and after the VVDS has been updated.

ISSUE-VCLOSE SCOPE

If you specify ISSUE-VCLOSE of BEFORE, AFTER, or YES, specify the scope of the modify command. Enter LOCAL to have the catalog modify command, F CATALOG,VCLOSE(targetvolser), issued only on the system that RENAME is running on. Enter SYSPLEX to have the catalog modify command, F CATALOG,VCLOSE(targetvolser), issued on the local system; the modify command will also be routed to all the other systems in the sysplex, via an MVS ROUTE *OTHER command, after the VVDS has been updated.

MAX-TASKS

Specify the maximum number of subtasks to be used for volume processing in the RENAME step. The maximum allowed value is 255. At some point, increasing the number of subtasks will cease to increase performance, due to resource contention. Specifying a value that is too large may result in termination due to memory constraints.

MISSINGUCAT

Specify the disposition of target volume data sets where the VVDS catalog back-pointer is not a catalog in the list supplied to the COPY step.

MISSINGUCAT RC

Specify the return code to be generated for the RENAME command if one or more target volume data sets contain a VVDS catalog back-pointer not in the list supplied to the COPY step.

NOTRENAMED

Specify the disposition of target volume data sets that are not renamed because they do not match a rename mask.

NOTRENAMED RC

Specify the return code to be generated for the RENAME command if one or more target volume data sets are not renamed because they do not match a rename mask.

ORPHANCATENTRY

Specify the disposition of target volume data set catalog entries where in some circumstances the data set is not found on the volume.

ORPHANCATENTRY RC

Specify the return code to be generated for the RENAME command if one or more target volume data set catalog entries do not have a corresponding volume data set.

RECATALOG

Specify YES to replace catalog entries encountered when cataloging target volume data sets.

RENAME-AUDIT-LOG

Enter YES to specify that an audit log of the data sets being renamed is to be created by RENAME volume processing.

RENAME-AUDIT-LOG SMF

If you specified YES in the RENAME-AUDIT-LOG field, enter the record type in this field. Valid values are 128 through 255 inclusive. SMF must be recording the specified record type. The layout of the records written can be found in member CKZRNSMF of the DB2 Cloning Tool JCL library.

RENAME-ERROR

This option specifies how processing proceeds when a RENAME error is encountered. Enter ABORT to terminate with an RC=8 after the first error to preserve integrity. ABORT is recommended. Enter CONTINUE to continue processing after most errors; the RENAME command will complete with the specified return code unless an error not handled by the CONTINUE logic is encountered.

CAUTION:

The use of CONTINUE can cause inconsistencies between the contents of the volumes and catalogs. If CONTINUE is specified, DB2 Cloning Tool will not guarantee integrity and the given results will not be fixed by DB2 Cloning Tool.

RENAME-ERROR CONTINUE RC

Specify the return code to be used if RENAME-ERROR CONTINUE is specified.

RENAME-LIST

Specify whether a list of the renamed data sets is to be produced by RENAME volume processing.

RENAME TYPE

Enter SAFE to allow a rerun of the RENAME command by backing up critical volume structures that are changed during the volume processing - the VTOC, VTOCIX, and VVDS. This backup data is used on a rerun to restore any volumes changed to the state they were in following the COPY step. This adds some slight execution time for RENAME to capture the portions of target volumes modified by RENAME. Incorrect rename masks may be a reason for needing to rerun the RENAME step. If multiple and complicated masks are required, this option is recommended. Also affecting the renaming is whether the data set naming conventions used by the application are fairly static or subject to frequent change - inferring that rename masks need to be watched.

To specify SAFE, you must also specify a DD name in the the VOLBKUP-DDN field. The VOLBKUP data set must not be deleted before a rerun of RENAME. If the VOLBKUP data set is lost, the COPY step will need to be run again, provided that the opportunity for correct point-in-time images has not been lost.

If source volume access is not resumed until the entire process is complete (implying that the same point-in-time images can be re-copied), the time to rerun the COPY step may be insignificant compared to adding some overhead with the SAFE option for every cycle.

SPEED is the opposite of SAFE. The RERUN option for the RENAME step will be rejected if attempted. Correction of any errors will require the COPY and RENAME step to be run again.

TEMPDSN

Specify the disposition of temporary data sets found on target volumes.

TEMPDSN RC

Specify the return code to be generated for the RENAME command if one or more temporary data sets are found on target volumes.

UPDATE-IAM-ASSOCIATIONS

Specify whether IAM data set associations are to be updated as part of RENAME processing. IAM must be active on the system for the updates to happen. This option addresses the situation where there are IAM data sets that are being cloned that include AIXes and PATHs, and it is desired to update the associations to correspond with the new data set names. The association information for IAM data sets will be determined and updated by internally using IDCAMS LISTCAT and IDCAMS DEFINE RECATALOG commands.

VALIDATE-SMS-CLASSES

Specify whether the SMS class names specified in the DATACLAS, DATACLAS-PAIRS, MGMTCLAS, MGMTCLAS-PAIRS, STORCLAS, and STORCLAS-PAIRS keywords will be validated as being defined to SMS (YES) or not (NO). This option addresses the situation where the target SMS class names are not defined on the system where RENAME is run.

VOLBKUP-DDN

Specify the DD name for the backup data set to be used for backing up target volume VTOCs, VTOCIXs, and VVDSs, to be used in the event of a rerun of the RENAME step. You must also enter SAFE in the RENAME TYPE field to use this parameter.

DB2FIX command defaults

DB2FIX will fix target DB2 page spaces that have LPL or GRECP status by issuing a DB2 START DATABASE command against them.

The following values can be set on the subsystem cloning DB2FIX command defaults panel. These values are used strictly as defaults for DB2FIX command parameters when you are creating a new subsystem cloning profile. The parameter values can later be edited in the profile.

DSNDB01-DBD01-STARTED RC

Specify the return code that will be used if table space DBD01 in database DSNDB01 was started due to the table space having LPL or GRECP status.

MAX-CONCURRENT-CMDS

Specify the maximum number of start commands that will be processed concurrently. Increasing this number can decrease the elapsed time of DB2FIX and will increase the resources being used by DB2. Experiment with this parameter to determine the optimum setting.

MEMBERS-AND-DBD01 RC

Enter the return code that will be used if table space DBD01 in database DSNDB01 was started (due to the table space having LPL or GRECP status), and if other members in this data sharing group need to be started because they hold locks.

MEMBERS-NEED-STARTING RC

Specify the return code that will be used and the action to be taken if other members in this data-sharing group need to be started because they hold locks. The default ACTION parameter for this keyword is ACTION(QUIT); to specify ACTION(CONTINUE), add the parameter to the generated JCL. Refer to the "DB2FIX" on page 379 command information for more information about the ACTION parameter.

START-SCOPE

Specify the scope of the DB2 start database commands that will be used. Enter DATABASE to specify that a single DB2 start database command will be done for all page spaces in a database that have LPL or GRECP status (e.g. STA DB(dbname) SPACENAM(*)). Enter PAGESPACE to specify that a single DB2 start database command will be done for each page space in the database that has LPL or GRECP status (e.g. STA DB(dbname) SPACENAM(spacename)).

WAIT ACTION

Specify the action to take when a timeout occurs. Enter QUIT to quit processing or CONTINUE to continue processing.

WAIT RC

Enter the return code to be used if the specified time limit expires before the DB2 start database command has fixed the page spaces in LPL or GRECP status. DB2FIX will terminate with this return code.

WAIT TIME

Specify the maximum time in minutes that DB2FIX will wait for a single DB2 start database command to fix the page spaces that were in LPL or GRECP status.

WAIT-AND-DBD01 RC

Enter the return code that will be used if table space DBD01 in database DSNDB01 was started due to its being found to have LPL or GRECP status and if the WAIT time limit was exceeded for a DB2 start database command to fix page spaces in LPL or GRECP status.

DB2SQL command defaults

The DB2SQL command generates and executes the SQL statements necessary to update the DB2 catalog.

The following values can be set on the subsystem cloning DB2SQL command defaults panel. These values are used strictly as defaults for DB2SQL command parameters when you are creating a new subsystem cloning profile. The parameter values can later be edited in the profile.

LISTSQL

Enter YES in this field to include the generated SQL in the listing.

WLM-ENV-NOT-UPDATED RC

Enter the return code to be used if there is a WLM ENVIRONMENT value in SYSIBM.SYSROUTINES that is not updated.

DATACLAS-NOT-UPDATED RC

Enter the return code to be used if there is a DATACLAS value in SYSIBM.SYSSTOGROUP that is not updated.

MGMTCLAS-NOT-UPDATED RC

Enter the return code to be used if there is a MGMTCLAS value in SYSIBM.SYSSTOGROUP that is not updated.

STORCLAS-NOT-UPDATED RC

Enter the return code to be used if there is a STORCLAS value in SYSIBM.SYSSTOGROUP that is not updated.

DB2START command defaults

The DB2START command is used to start a DB2 subsystem, via a z/OS **START DB2** command, as part of cloning a DB2 subsystem.

The following values can be set on the subsystem cloning DB2START command defaults panel. These values are used strictly as defaults for DB2START command parameters when you are creating a new subsystem cloning profile. The parameter values can later be edited in the profile.

DB2-ALREADY-RUNNING RC

Specify the return code that will be used if the DB2 subsystem is already running.

WAIT time

Enter the maximum time in minutes that DB2START should wait for the DB2 subsystem to start.

WAIT RC

Enter the return code to be used if the specified time limit expires before the DB2 subsystem has started.

DB2STOP command defaults

The DB2STOP command is used to stop a DB2 subsystem, via the DB2 **STOP DB2** command, as part of cloning a DB2 subsystem.

The following values can be set on the subsystem cloning DB2STOP command defaults panel. These values are used strictly as defaults for DB2STOP command parameters when you are creating a new subsystem cloning profile. The parameter values can later be edited in the profile.

CASTOUT

Enter the CASTOUT value that will be used in the STOP DB2 command.

DB2-ALREADY-STOPPED RC

Specify the return code to be used if the DB2 subsystem is already stopped.

MODE

Specify the MODE value that will be used in the STOP DB2 command. Refer to the DB2 command reference for the meanings of the FORCE and QUIESCE keywords and their values.

WAIT time

Specify the maximum time in minutes that DB2STOP should wait for the DB2 subsystem to stop.

WAIT RC

Specify the return code to be used if the specified time limit expires before the DB2 subsystem has stopped.

DB2UPDATE command defaults

DB2UPDATE may be used to make some of the changes within DB2 to reflect the data sets that are renamed during a subsystem cloning.

The following values can be set on the subsystem cloning DB2UPDATE command defaults panel. These values are used strictly as defaults for DB2UPDATE command parameters when you are creating a new subsystem cloning profile. The parameter values can later be edited in the profile.

DB2-XCFCLEAN

Enter YES to specify that the target DB2 data-sharing group XCF structures and group members should be cleaned up.

DDF-NOT-UPDATED RC

Specify the return code to be used if there is a DDF record in the BSDS but the DDF values were not completed on the Enter DB2 DDF values panel.

HLQ-NOT-UPDATED RC

Specify the return code to be used if the VSAM catalog name in the BSDS is not updated or a VCAT in the DB2 directory table space DBD01 is not updated. If you do not change the VSAM catalog name in the BSDS, or a VCAT in the DB2 directory table space DBD01, the target DB2 system may access table and index spaces on the source DB2 system.

DB2UTILXCLEAN command defaults

DB2UTILXCLEAN removes all entries from SYSUTILX. This command should be specified when utilities might be running or registered in SYSUTILX when the source DB2 subsystem is cloned.

The following values can be set on the subsystem cloning DB2UTILXCLEAN command defaults panel. These values are used strictly as defaults for DB2UTILXCLEAN command parameters when you are creating a new subsystem cloning profile. The parameter values can later be edited in the profile.

Clean SYSUTILX

Enter YES in this field if utilities might be running or registered in SYSUTILX when the source DB2 subsystem is cloned. If SYSUTILX is not cleaned out, the source DB2 subsystem can become corrupted when the target DB2 subsystem cleans up the entries in SYSUTILX with a -TERM UTIL command. The DB2UTILXCLEAN command cleans out the target SYSUTILX and its indexes.

If utilities were running or registered in SYSUTILX when the source DB2 subsystem was cloned, target objects might be in UT status after the cloning. These objects can be identified and the UT status can be removed by using the sample REXX exec provided in the CKZDUTCL member of SCKZJCL.

RESTORE-FROM-DUMPTAPES command defaults

RESTORE-FROM-DUMPTAPES restores system-level backup tapes to the target volumes.

The following values can be set on the subsystem cloning RESTORE-FROM-DUMPTAPES command defaults panel. These values are used strictly as defaults for RESTORE-FROM-DUMPTAPES command parameters when you are creating a new subsystem cloning profile. The parameter values can later be edited in the profile.

BACKINFO ddname

Enter the DD name that points to the data set containing the backinfo data.

VOLPAIRS ddname

Enter the DD name that points to the VOLPAIRS output data set.

USERCATALOGS ddname

Enter the DD name that points to the UCATs output data set.

STATUS ddname

Enter the DD name that points to a STATUS output data set. This data set is provided to allow resuming the job.

Max Tape Drives

Enter the maximum number of tape drives available for the RESTORE-FROM-DUMPTAPES process. Because each restore job uses one tape drive and runs as a separate subtask, this is also the number of restore subtasks.

Tape Unit

Enter the device name to use for the tape allocation.

Vary Scope

Specify the scope of VARY commands for target volumes. Enter GLOBAL to issue the commands across the sysplex. Enter LOCAL to issue VARY commands only across the local system (the same system that the job runs on). Note: Specifying LOCAL might cause resource conflicts with the RESTORE-FROM-DUMPTAPES job or other jobs running on other systems.

Setting table space cloning defaults

This topic describes how to specify defaults for table space cloning commands, DDs, and other defaults.

Defaults are originally derived from the CKZINI PARMLIB member, but can be customized for each user ID using these panels. These defaults are then saved in ISPF profiles and are specific to each TSO user ID. This allows different users to have different defaults when creating cloning profiles.

Enter 2 on the User Settings menu to access table space cloning defaults. The User tablespace clone settings panel is displayed:

```

User DB2 tablespace clone settings
Option ==>
1 DD Specification
2 SET Command
3 COPY Command
4 HLQDDDF Command
5 XML Object Definition

```

Command defaults

To set default values for each command, enter the appropriate number for the command in the **Option** field.

Setting default DD specifications for table space cloning

The DB2 tablespace clone DD Specification panel lets you enter default DD specifications for the DDs required for table space cloning and for optional user DDs. These DDs may be used in one or more of the source, target, or TCPIP jobs.

Note: You must allocate any data sets that do not exist before attempting to execute the table space cloning jobs.

DD defaults can be customized for each user ID using these panels. These defaults are then saved in ISPF profiles and are specific to each TSO user ID. This allows different users to have different defaults when creating cloning profiles.

Enter **1** on the User DB2 tablespace clone settings menu. The DB2 tablespace clone DD Specification panel is displayed:

```

DB2 tablespace clone DD Specification
Command ==>                               Scroll ==> PAGE
Commands:      D - Set Defaults  C - Clear Defaults  U - User DD Specification
Line commands: S - Select/Unselect

Control DD defaults:
HLQ . . . . TWUSR                          (control HLQ)
Member . . . LSTDMBR                        (control member)
                                           Row 1 of 19

  DD Name  DD
SEL CKZIN  *
SEL CKZPRINT  SYSOUT=*
SEL CKZINI   DISP=SHR,DSN=CKZ.WRK0310.SCKZPARM(CKZINI)
SEL CKZLOG   SYSOUT=*
SEL CKZLSTDF DISP=SHR,DSN=TWUSR.LISTDEF(LSTDMBR)
           CKZMSKDF DISP=SHR,DSN=TWUSR.MASKDEF(LSTDMBR)
           CKZCRXML DISP=SHR,DSN=TWUSR.XMLCRDDL(LSTDMBR)
SEL CKZS    DISP=OLD,DSN=TWUSR.SYNCDDB2(LSTDMBR)
           CKZC    DISP=OLD,DSN=TWUSR.COPYDSNS(LSTDMBR)
           CKZM    DISP=OLD,DSN=TWUSR.XMLSTR(LSTDMBR)
SEL CKZQ    DISP=OLD,DSN=TWUSR.SQLOUT(LSTDMBR)
           CKZW    DISP=OLD,DSN=TWUSR.CMDSSTPT(LSTDMBR)
           CKZX    DISP=OLD,DSN=TWUSR.CMDSSTPS(LSTDMBR)
           CKZY    DISP=OLD,DSN=TWUSR.CMDSSTRS(LSTDMBR)
           CKZZ    DISP=OLD,DSN=TWUSR.IDCAMS(LSTDMBR)
           CKZD    DISP=OLD,DSN=TWUSR.DDLOUT(LSTDMBR)
           CKZRRJOB DISP=OLD,DSN=TWUSR.RRJOB
           CKZRRDSN DISP=OLD,DSN=TWUSR.RRDSN
SEL CKZERROR  SYSOUT=*
           SYSINCKZ DISP=SHR,DSN=TWUSR.LOGAPCTL(LSTDMBR)
           SYSOUT  SYSOUT=*

```

Required DD names are pre-selected with SEL next to of the DD name and will be included in one or more of the three DB2 Cloning Tool Table Space Cloning jobs (source, target, or TCP/IP server).

For most DD names, you can modify the data set specifications such as DISP or the SYSOUT location. Some DD names are not required, but when specified must have a particular DD name. These DD names cannot be modified.

Refer to the information that follows to determine which DDs to include in your table space cloning jobs.

Editing DD names and specifications

Where allowed, you can change the DD names and specifications to meet your site's requirements. You can type directly over the current values in the fields.

Using the Control DD defaults fields to edit DD specifications

The **Control DD** fields can be used to easily set the high level qualifiers and member names for all the DDs on the panel that can be modified.

1. Specify a default high level qualifier in the **HLQ** field and a default member name in the **Member** field.
2. Enter D in the **Command** field
3. When you press Enter, the DD name fields are populated with the specified HLQ and member name.

To clear the DD specifications, enter C in the **Command** field.

Attention: if you use the C command, all specifications that will be modified will be cleared.

Selecting or deselecting a DD for inclusion in JCL

Use the S line command as a toggle to select or deselect a DD. If the DD will be included, SEL is displayed next to the DD name. If a DD is not selected, it will not be included by default in the table space cloning profile, but can be added later when creating a table space cloning profile.

Adding user DDs

You can enter your own user-defined DD names and specifications on a separate panel. Enter the U command in the **Command** field. When you press Enter, the following panel is displayed:

```
DB2 tablespace clone DD Specification          Scroll ==> PAGE
Command ==>
Commands:      A - Add Line  P - Product DD Specification
Line commands: S - Select/Unselect  D - Delete Line

DD Name DD
```

This panel allows you to enter default DD specifications for user DDs that you may want to include in table space cloning jobs.

To add DDs to this panel:

1. Enter A in the Command line and press Enter.
2. In the lines that are displayed in the input area, enter the DD name and desired data set specifications.
3. Enter S next to the DD name to select the DD for inclusion in jobs.

Use the S line command as a toggle to select or deselect a DD for inclusion. If a DD is not selected, it will not be included by default in the table space cloning profile, but can be added later when creating a table space cloning profile.

To remove a DD, enter D next to the DD and press Enter.

To return to specifying table space cloning DDs, enter P in the command line.

Table space cloning DD descriptions:

The table space cloning DDs provided by DB2 Cloning Tool Table Space Cloning on the DB2 tablespace clone DD Specification panel are identified in this topic.

The following table describes the table space cloning DDs.

Table 45. Table space cloning DD descriptions on the DB2 tablespace clone DD Specification panel

DD	Required?	Usage	Default
CKZIN	Yes	In the source job, it identifies source (local) and target DB2 subsystem names, identifies DDs passed to ADRDSSU for data set allocations, and contains other commands related to the source and target subsystems. The CKZIN DD also contains the COPY command and its options, and various SET commands.	*
		In the target job, the CKZIN DD contains the input parameters for the target job. It must point to the data set referenced by the SYNCDB2-DDN subcommand in the source job COPY command.	
		In the TCP/IP server job, the CKZIN DD contains the input parameters for the TCP/IP job in the form of SET commands.	
CKZPRINT	Yes	In the source job, CKZPRINT displays CKZINI tokens, control parameters, data set names and associated DB2 table spaces and index spaces, DB2 start and stop space command status, and DFSMSdss program ADRDSSU commands and status.	SYSOUT=*
		In the target job, CKZPRINT displays CKZINI tokens, CKZIN control parameters, DB2 SQL execution status, and SYNCDB2 status and START DB2 command status for each data set processed.	
CKZINI	Yes	CKZINI is the product PARMLIB member that is set up during installation. Used in the source and the target jobs, the CKZINI member contains program variables.	The PARMLIB library where the PARMLIB member CKZINI is located.

Table 45. Table space cloning DD descriptions on the DB2 tablespace clone DD Specification panel (continued)

DD	Required?	Usage	Default
CKZLOG	No	In the source job, CKZLOG displays LISTDEF processing, DB2 commands issued, and the responses or results of the commands.	SYSOUT=*
		In the target job, CKZLOG displays the DB2 commands issued and the responses or results of the commands. It also displays detailed information about each DB2 page access.	
CKZLSTDF	Yes	In the source job, CKZLSTDF contains LISTDEF-like commands with standard IBM syntax to select the source table spaces and index spaces to be processed and, if requested, copied to the target.	DISP=SHR,DSN=h1q-field. LISTDEF(member-field)
CKZMSKDF	Yes, if data masking will be used	CKZMSKDF is used in the source job to hold the masking rules to be applied and the tables to be masked. It is passed to the target job as an input to the data masking processor.	DISP=SHR,DSN=h1q-field. MASKDEF(member-field)
CKZCRXML	Yes if cloning tables that contain XML data	CKZCRXML is not used in the source job. In the target job, this DD contains the DDL to create an XML table for adding strings to SYSIBM.SYSXMLSTRINGS using an SQL INSERT. The data set pointed to by the CKZCRXML DD must have an LRECL of 80 and RECFM of FB.	DISP=SHR,DSN=h1q-field. XMLCRDDL(member-field)
CKZS	Yes in most cases	In the source job, CKZS is an output data set that will be used by the target job to make the VSAM objects accessible on the target DB2 subsystem.	DISP=OLD,DSN=h1q-field. SYNCDB2(member-field)
CKZC	No	In the source job, CKZC points to a data set which will contain a list of TO and FROM data set names derived from the LISTDEF command input. This DD is provided to assist in copying the selected VSAM objects outside of DB2 Cloning Tool Table Space Cloning and is only used if the keyword for the COPY command is DATA-MOVER(PGM(NONE)).	DISP=OLD,DSN=h1q-field. COPYDSNS(member-field)
CKZM	Yes if cloning DB2 tables containing XML data	In the source and target jobs, CKZM is used to pass the source subsystem XMLSTRINGS catalog table contents from the source job to the target job.	DISP=OLD,DSN=h1q-field. XMLSTR(member-field)
CKZQ	Yes if SQL is generated by the source job	In the source and target jobs, CKZQ is used to pass SQL commands to the target job to synchronize identity column values between the target and source.	DISP=OLD,DSN=h1q-field. SQLOUT(member-field)
CKZW	No	In the source job, CKZW is used to submit DB2 commands to stop the target DB2 table spaces and index spaces and to deallocate the target VSAM data sets.	DISP=OLD,DSN=h1q-field. CMDSSSTPT(member-field)
CKZX	No	In the source job, CKZX is used to submit DB2 commands to stop the source DB2 table spaces and index spaces and to deallocate the source VSAM data sets.	DISP=OLD,DSN=h1q-field. CMDSSSTPS(member-field)
CKZY	No	In the source job, CKZY is used to submit DB2 commands to start the source DB2 table spaces and index spaces after they have been copied to the target.	DISP=OLD,DSN=h1q-field. CMDSSSTRS(member-field)

Table 45. Table space cloning DD descriptions on the DB2 tablespace clone DD Specification panel (continued)

DD	Required?	Usage	Default
CKZZ	No	In the source job, CKZZ is used to output IDCAMS deletes and renames for all data sets that were copied as .F0001 data sets, rather than .I0001 or .J0001. This might occur if some or all target table spaces or index spaces did not exist in the DB2 catalog before the source job is run. This job can then be used as input to the CKZFIX job to correct missing target data sets.	DISP=OLD,DSN=h1q-field. IDCAMS(member-field)
CKZD	Yes if DDL for creating non-existent target objects will be generated or executed	The CKZD is used in the source job. Depending on the type of DDL processing specified, it contains source or target object DDL.	DISP=OLD,DSN=h1q-field. DDLOUT(member-field)
CKZRRJOB	No	If using the target job runtime repository for rerunning the target job or keeping a job history, this DD will be included in the source, target, and report jobs.	DISP=OLD,DSN=h1q-field.RRJOB
CKZRRDSN	No	If using the target job runtime repository for rerunning the target job or keeping a job history, this DD will be included in the source, target, and report jobs.	DISP=OLD,DSN=h1q-field.RRDSN
CKZERROR	No	When CKZERROR is included in source, target, and/or TCP/IP server jobs, all warning and error messages are output to this DD, as well as to CKZPRINT.	SYSOUT=*
SYSINCKZ	Yes if image copies will be used as the source for the cloning	This DD is used for log apply processing when image copies are used as the source for the table space cloning.	DISP=SHR,DSN=h1q-field. LOGAPCTL(LSTDMBR)
SYSOUT	Yes if image copies will be used as the source for the cloning	This DD is used for message output when image copies are used as the source for the table space cloning.	SYSOUT=*

SET command defaults

The SET command specifies the local DB2 subsystem and other job-wide specifications for each of the three jobs: source, target and TCP/IP server.

The following values can be set on the table space cloning SET command defaults panel. These values are used strictly as defaults for SET command parameters when you are creating a new table space cloning profile. The parameter values can later be edited in the profile.

LOCAL-SSID

Specify the name of the local DB2 subsystem. This is the source subsystem for the source job and the target subsystem for the target and TCPIP server jobs. To display a list of subsystems to choose from, enter an asterisk (*) in this field and press Enter.

| **SOURCE-IP-VERSION6**

| Indicate the version of TCP/IP used on the source TCP/IP server. Enter
| YES for IPv6 and NO for IPv4.

| **SOURCE-TCPIP-SERVER-IP**

| Enter the IP of the LPAR on which the source TCP/IP server job runs. This
| job is required for cross-LPAR log apply functionality.

| **SOURCE-TCPIP-SERVER-PORT**

| Enter the port that the source TCP/IP server job listens on for client
| connects. This is the port used by the target job to connect to the source
| server when using cross-LPAR log apply functionality.

| **SOURCE-TCPIP-STC-NAME**

| Enter the source TCP/IP server job started task name.

| **IP-VERSION6**

| Indicate the version of TCP/IP used on the TCP/IP server. Enter YES for
| IPv6 and NO for IPv4.

| **TCPIP-SERVER-IP**

| Enter the IP of the LPAR on which the TCP/IP server job runs.

| **TCPIP-SERVER-PORT**

| Enter the port that the TCP/IP server job listens on for client connects.

| **TCPIP-STC-NAME**

| Enter the local TCP/IP started task. TCPIP is the default.

| **ADVISORY-STATUS-VALUES**

| Enter YES in this field to check the status of table spaces and index spaces
| before copies are performed. During cloning, if a specified advisory status
| is detected, the space is marked mismatched and a warning message is
| issued. If ALLOW-COPY-ON-MISMATCH(YES) and MAX-RC(4) are in
| effect, the copy may proceed; otherwise the copy is not allowed for all
| affected data sets. To specify values for the ADVISORY-STATUS-VALUES
| parameter, enter A in the Option line and press Enter.

| **DEFAULT-SQLID**

| If you want to apply a creator ID to all tables and indexes in LISTDEF
| without an explicit creator ID, enter the default SQL ID in this field.

| **DB2-COMMAND-RESPONSE-WAIT(secs)**

| Enter the number of seconds to wait for a space to go from STOPP to
| STOP status after the STOP DB2 command is issued.

| **DB2-PLAN**

| Enter the product plan name that will be used to issue SQL against the
| DB2 catalog on the source and the target subsystem, and to submit
| dynamic SQL for the target subsystem. The plan name must be the same
| for all three jobs: source, target, and TCPIP server.

| **MAX-COPY-RC**

| Enter the return code that, when exceeded, will cause a job to stop copying
| of data. This field only applies to copy processing return codes. This allows
| one or more copies to fail and the others to continue.

| **MAX-RC**

| Enter the return code that, when exceeded, causes a job to stop. This
| excludes data set copy processing (see the MAX-COPY-RC field). If some
| table spaces or index spaces are not found on the target;
| COPY-IF-NO-DB2-TARGET-OBJECTS is N; and MAX-RC is set to 0, no

table spaces or index spaces will be copied. Set MAX-RC to 4 if you want to complete the cloning process for table spaces and index spaces that can be copied, even if the remainder of the table spaces or index spaces cannot be cloned.

MAX-SUBTASKS

Specify the number of subtasks that should be used. Subtasks are used to perform several functions, such as catalog access in the source job and SYNCDB2 command processing in the target job. These subtasks allow multiple I/Os to be performed concurrently. Changing the number of subtasks may improve performance.

SUBTASK-DATASET-EXTENSIONS

Specify Y in this field to allow data set extents to be processed in any available subtask. When a significant number of extents may be processed by the target job, setting this parameter to Y might improve the total elapsed time of the target job. If DB2 START commands are enabled for target objects, the objects will be started when the last extent of an object has been processed, regardless of the subtask used. The number of subtasks (set via PARMLIB parameter MAX_SUBTASKS or COPY command MAX-SUBTASKS parameter) must be greater than one for this to be a useful feature.

SUBTASK-TERMINATION-WAIT

Specify a wait time for all subtasks to end before exiting. In the target job, a large data set being updated with new OBIDs can take a long time to process. In this case, DB2 Cloning Tool may exit before the subtask ends, resulting in an A03 ABEND. Valid values are 0 to 9999 seconds; 0 indicates that DB2 Cloning Tool is to wait for all subtasks to end before exiting.

MERGE-PRINT

Enter YES in this field to specify that message output to CKZ PRINT and CKZ LOG are to be combined into CKZ PRINT. This might be useful when investigating a problem or sending documentation to the support center.

RESTRICT-STATUS-VALUES

Enter YES in this field to check the status of table spaces and index spaces before copies are performed. During cloning, if a specified restricted status is detected, the space is marked mismatched and a warning message is issued. If ALLOW-COPY-ON-MISMATCH(YES) and MAX-RC(4) are in effect, the copy may proceed; otherwise the copy is not allowed for all affected data sets. To specify values for the RESTRICT-STATUS-VALUES parameter, enter R in the Option line and press Enter.

TEMPLATE-VARIABLE

Enter YES in this field to use job template variables. To specify the variables, enter T in the Option line and press Enter.

REMOTE-CONNECT-TYPE

Specify a particular connection type for the remote (target) subsystem. Specify C to indicate CAF, D for DDF, and T for TCPIP, or blank.

When this field is left blank, connections are attempted in this order: CAF, DDF, and TCPIP. This setting is the preferred method of specifying all DDF and TCPIP connections. For example, if this field is left blank when TCPIP is used and DDF fails, error messages are printed for the DDF failure. If REMOTE-CONNECT-TYPE is set to T, then DDF is not attempted and thus no error message will be issued.

When the source and target DB2 subsystems have the same name, specify T to indicate that the same named subsystem is used for both source and target. This allows the target subsystem to be connected via TCPIP.

USE-RUNTIME-REPOSITORY

Specify YES in this field to indicate that when the target job is executed, it is to only process the data sets that have not been successfully processed. This restart process is enabled by a DB2 Cloning Tool repository. The repository keeps track of target jobs and all the data sets that are processed by the target job. This allows the failed target job to be restarted, skipping successfully processed target data sets. Data sets in progress are not restarted.

TARGET-JOB-REPAIR-EXECUTE

Specify YES in this field to generate REPAIR jobs to detect and correct catalog or DB2 version inconsistencies. This can occur when DB2 objects are copied from one subsystem to another. If you specify YES in this field, REPAIR jobs are submitted near the end of the target job. If you specify NO, no REPAIR jobs are executed in the target job. When TARGET-JOB-REPAIR-EXECUTE and TARGET-JOB-REPAIR-SELECT are both YES and a table space is selected via selection criteria, a REPAIR job is executed for that table space or index space.

The inconsistencies processed are record format (BRF vs. RRF), actual page format vs. catalog column RBA_FORMAT, HASHDATAPAGES, and versioned objects. Record format, page format, and HASHDATAPAGES processing only apply to DB2 V11 and may only be run on table spaces. Versioned object processing applies to all DB2 versions; DB2 V9 and V10 REPAIR VERSIONS can be run on table spaces and index spaces, while DB2 V11 REPAIR VERSIONS can be run on table spaces only.

TARGET-JOB-REPAIR-SELECT

If you specify YES in this field, the REPAIR job is processed at the end of the target job only for those table spaces or index spaces matching the criteria defined for the DB2 version being processed. If you specify NO, no table spaces or index spaces are processed.

TARGET-JOB-REPAIR-TEST

If you specify YES in this field, the REPAIR job is submitted, but the mismatch information will not be corrected in the catalog. The mismatch information that results in non-zero return codes is reported in the CKZPRINT output DD.

REBUILD-INDEXES-EXECUTE

Specify YES in this field to submit index rebuilds as part of the target job. If you specify YES in this field, you must provide a DD name in the COPY command TARGET-JOB-INDEX-REBUILD-DDN field.

Specifying ADVISORY-STATUS-VALUES defaults:

When you specify YES for ADVISORY-STATUS-VALUES field on DB2 tablespace clone SET Command panel, a DISPLAY DATABASE command is issued for each database on the source and target subsystem using the keywords that you specify. The statuses are included as part of the SET command. You must specify the specific status values you want to include.

Enter A in the **Option** field on the DB2 tablespace clone SET Command panel. The Specify Advisory Status Values panel is displayed. This panel allows you to enter default settings for the table space cloning SET command ADVISORY-STATUS-VALUES keyword.

The possible advisory status values are provided in the following list. A short description of each status is displayed to the right of the advisory status abbreviation on the panel. Refer to the DB2 utilities documentation for your version of DB2 for detailed information about the statuses.

ARBDP

Advisory REBUILD-pending

AREO*

Advisory REORG-pending

AREOR

Advisory REORG-pending (DB2 10 only)

AUXW

Auxiliary warning

ICOPY

Informational COPY-pending

Selecting or deselecting an advisory status

Select or deselect a specific advisory status value by entering S in the line command area next to the advisory status. Enter one of the following in the **Command** field to select multiple statuses:

- Enter A to select all advisory status values.
- Enter C to clear all the selected advisory status values.
- Enter D to select only the default advisory status values. The default statuses are ARBDP, AREO*, AREOR (for DB2 10 and later), and AUXW.

Specifying RESTRICT-STATUS-VALUES defaults:

When you specify YES for the RESTRICT-STATUS-VALUES field on DB2 tablespace clone SET Command panel, a DISPLAY DATABASE command is issued for each database on the source and target subsystem using the RESTRICT keywords that you specify. The statuses are included as part of the SET command. You must specify the specific status values you want to include.

Enter R in the **Option** field on the DB2 tablespace clone SET Command panel. the Specify Restrict Status Values panel is displayed. This panel allows you to enter default settings for the table space cloning SET command RESTRICT-STATUS-VALUES keyword.

The possible restricted status values are provided in the following list. A short description of each status is displayed to the right of the advisory status abbreviation on the panel. Refer to the DB2 utilities documentation for your version of DB2 for detailed information about the statuses.

ACHKP

Auxiliary CHECK-pending

CHKP CHECK-pending

COPY COPY-pending

GRECP Group buffer pool RECOVER-pending

LPL Logical page list entries

RBDP REBUILD-pending

RECP RECOVER-pending

REORP REORG-pending

RO Read-only mode

STOP Stopped objects, including the restricted states STOP, STOPE, STOPP, and LSTOP

UT Utility access mode

UTRO Serialized for utility access and available for read-only access

UTRW Serialized for utility access and available for read-write access

UTUT Serialized for utility access and unavailable

UT* Any utility access mode: UT, UTRW, UTRO, or UTUT

WEPR Displays write error page range information

Selecting or deselecting a restricted status

Select or deselect a specific restricted status value by entering S in the line command area next to the status. Enter one of the following in the **Command** field to select multiple statuses:

- Enter A to select all status values.
- Enter C to clear all the selected status values.
- Enter D to select only the default status values. The default restricted statuses are ACHKP, CHKP, GRECP, RBDP, RECP, REORP, and UT*.

Specifying default job template variables:

When you specify YES for the TEMPLATE-VARIABLE field on DB2 tablespace clone SET Command panel, job template variables are included in the table space cloning jobs. You must define the variables to be used in the templates.

Enter T in the **Option** field on the DB2 tablespace clone SET Command panel. The Specify Job Template Variables panel is displayed. Use this panel to define your own variables to be used in a job template. Job templates consist of the z/OS JCL statements, DSS commands, user variables, and processing variables that are used for input.

Detailed instructions about setting up job templates and variable descriptions can be found in the sample members CKZJOB1, CKZJOB2 and CKZJOB3, provided in the product sample library.

Name Enter the variable name.

Variable

Enter the value for the variable.

| **Note:** The following variable names are reserved and may not be specified as a
| user variable on this panel:

- | • JOBCARD
- | • DATABASE
- | • TABLESP
- | • TABLE
- | • INDEXSP
- | • INDEX
- | • CREATOR
- | • SRCSSID
- | • TRGSSID
- | • SRCOBS
- | • TRGOBS

Selecting or deselecting a variable for inclusion in JCL

Use the S line command as a toggle to select or deselect a variable to be included by default. If the variable will be included, SEL is displayed next to the variable name. If a variable is not selected, it will not be included by default in the table space cloning profile, but can be added later when creating a table space cloning profile.

You can also:

- Enter N in the Command field add a new line for a job template variable.
- To select all job template variables for inclusion, enter A in the Command field.
- To clear all selections, enter C in the Command field.

COPY command defaults

The COPY command controls all phases of replicating the DB2 table spaces and index spaces that are to be cloned.

The following values can be set on the table space cloning COPY command defaults panel. These values are used strictly as defaults for COPY command parameters when you are creating a new table space cloning profile. The parameter values can later be edited in the profile.

TARGET-DB2 SSID

Specify the default target DB2 subsystem. This field is required. To display a list of subsystems to choose from, enter an asterisk (*) in this field and press Enter.

LOCATION

Specify the DDF location name of the target DB2.

USERID

If required at your site, enter a user ID that may be needed to access DDF.

PASSWORD

If required at your site, enter a password that may be needed to access DDF.

SERVER-IP

| Specify the IP address of the LPAR on which the target DB2 subsystem
| resides.

SERVER-PORT

Specify the port that the TCP/IP server uses to wait for requests from the source job and that the source job uses to connect to the TCP/IP server job. The port assigned to the TCP/IP server cannot be a port used by DB2 or any other application.

DEFVCAT

Enter the default high level qualifier used for the target data sets if the target DB2 table spaces or index spaces are not predefined.

DATA-MOVER PGM

Enter the program to be used to initiate copies and copy options. Enter ADRDSSU to specify that COPY is to initiate FlashCopy or SnapShot under the covers via execution of DFSMSdss. Enter EMCAPI to invoke EMC TimeFinder/Clone Mainframe Snap Facility's data set level support. Enter SRCIMCPY to specify that image copies are to be used as the source for the table space cloning. Enter NONE to specify that no DATA-MOVER is to be invoked by COPY. NONE infers that data set copies will be created by you between the execution of the source job and the execution of the target job. When NONE is specified, COPY still captures necessary DB2 catalog information. NONE may also be used to verify object compatibility from source to target and to ensure parameters are correctly specified.

FASTREP

If you specify ADRDSSU in the DATA-MOVER PGM field, indicate whether fast replication is preferred (PREF), required (REQ), or not required (NONE). PREF is the default. Source and target pairs will be set up for a fast replication if PREF or REQ is specified. If NONE is specified, a 'normal' copy is allowed. If the level of ADRDSSU indicates it supports this keyword, the keyword will be passed to ADRDSSU.

FCTOPPRCPRIARY

If you specify ADRDSSU in the DATA-MOVER PGM field, specify YES in this field to indicate that a FlashCopy target volume can also be a PPRC primary volume. This applies to ESS devices only. This does not apply when NONE is specified in the FASTREP field.

If the target volume of the FlashCopy operation is a Metro Mirror primary device, you can enter one of the following to specify whether the device pair is allowed to go to duplex pending state:

- Enter PRESMIRREQ to specify that if the target volume is a Metro Mirror primary device, the pair must not go into a duplex pending state as the result of a FlashCopy operation.
- Enter PRESMIRRPREF to specify that if the target volume is a Metro Mirror primary device, it would be preferable that the pair does not go into a duplex pending state as the result of a FlashCopy operation. However, if a Preserve Mirror operation cannot be accomplished, the FlashCopy operation is still to be performed.
- Enter PRESMIRNONE to specify that a Preserve Mirror operation is not to be done, even if all of the configuration requirements for a Preserve Mirror operation are met. If the target specified is a Metro Mirror primary device, the pair is to go into a duplex pending state while the secondary device is updated with the tracks to be copied. PRESMIRNONE is the default if you specify FCTOPPRCPRIARY YES and the target is a Metro Mirror primary device.

When you specify FCTOPPRCPRIARY(YES) or FCTOPPRCPRIARY(PRESMIRNONE), the FlashCopy operation causes a

PPRC primary volume to become a FlashCopy target volume. A Metro Mirror or Global Copy pair currently in full duplex state goes into a duplex pending state when the FlashCopy relationship is established. When Metro Mirror or Global Copy completes the copy operation, the Metro Mirror or Global Copy pair goes to full duplex state. To prevent Metro Mirror or Global Copy pairs from going to duplex pending state during FlashCopy operation, you must specify FCTOPPRCPRIARY(PRESMIRREQ).

NULLSTORCLAS

If you specify ADRDSSU in the DATA-MOVER PGM field, you can use this field to indicate whether DFSMSdss is to pass a null storage class to the automatic class selection (ACS) routine. Enter YES to specify the null storage class. This can be useful if you are using non-SMS storage for VSAM object data sets. For more information on this keyword, consult the DFSMSdss Storage Administration documentation.

CMDDDDNAME

If you are planning to use a job template, specify the name of the output DD that will contain the job built using the JOB-TEMPLATE parameter of the COPY command. DB2 Cloning Tool Table Space Cloning will submit this set of commands in lieu of its internally built DSS commands. You only need to specify a DD name in this field if you are always planning to use a job template. Normal table space cloning processing occurs when a DD name is not specified. This option is only available for PGM(ADRDSSU).

PROCESS-DDL DDL-ENABLE

Enter YES in this field to generate or execute DDL to be used for creating nonexistent target objects.

PROCESS-TYPE

Indicate the type of DDL processing:

- Enter YES to generate and execute DDL for nonexistent target objects.
- Enter NO to not generate or execute DDL.
- Enter GEN to generate DDL for missing target objects, but do not execute the DDL.
- Enter EXEC to not generate DDL, but execute DDL already in the PROCESS-DDL-DDN field.
- Enter ALL to generate all source object DDL, but do not execute the DDL.

PROCESS-DDL-DDN

Specify the DD name where the generated DDL will be written to or read from. For DDL to be written to this DD, the DDL-ENABLE field must be set to YES, and the processing type must generate DDL (YES, GEN or ALL).

IGNORE-CREATE-OBJECT-EXISTS

Enter YES in this field to ignore -601 SQL errors that may occur when the DDL is executed. These errors are related to creating an object with the same name on the DB2 system. This field applies when PROCESS-TYPE is EXEC or YES.

GENERATE-DDL-DEFAULTS

Enter YES in this field to generate DDL statements for default values. If set to NO, the statements are omitted. For example, if this field is set to YES

and the catalog value for CLOSE is Y (the default), CLOSE YES is output in the generated DDL. If this field is set to NO, CLOSE YES is NOT included in the DDL.

INCLUDE DDL-ATTRIBUTE-CHANGE

Enter YES to include the DDL-ATTRIBUTE-CHANGE subcommand. If you specify YES, you must define the attribute changes on a separate panel. Enter D in the Command line to access the DB2 tablespace DDL-ATTRIBUTE-CHANGE Command panel.

ALLOW-COPY-ON-MISMATCH

Specify whether a copy should be allowed if one or more object attributes do not match between the source and target table space or index space. Enter YES to indicate that the copy should still be performed if the CKZINI PARMLIB parameter MAX_RC is set to 4 (or if the CKZINI PARMLIB parameter MAX_RC is set to 0 but MAX-RC on the SET command is set to 4). Enter NO to indicate that the table space or index space should not be copied and the job should complete with a return code of 4.

Tip: When using this keyword, set the PARMLIB keywords MAX_COPY_RC to 8 and the MAX_RC to 4 to allow copying of only the table spaces and index spaces that are safe to copy, i.e. table spaces and index spaces not in restrictive or advisory states. For all other source table spaces and index spaces, error messages will be produced and the table spaces and index spaces are not copied.

ALWAYS-COPY-HISTORY-TABLES

Specify YES to copy both history tables and non-history tables associated with all objects specified on a LISTDEF. When ALWAYS-COPY-HISTORY-TABLES is set to NO, history tables are not selected; however, history tables can be selected for copying using the HISTORY parameter on a LISTDEF command.

ALWAYS-COPY-INDEXSPACES

Enter YES to always copy index spaces. When set to YES, for every table space included in a LISTDEF, all index spaces are also included. No LISTDEF INCLUDE INDEXSPACES syntax is required.

TARGET-JOB-INDEX-REBUILD-DDN

This DD is used as part of the process to rebuild indexes. Any table in the target job that has a page changed via data masking or log apply requires index(es) to be rebuilt. To use this feature, specify a DD in this field and refer to the required additional instructions in the topics Chapter 18, "Using image copies to clone table spaces and index spaces," on page 259, Chapter 17, "Using data masking with table space cloning," on page 239, or Chapter 19, "Using LOG-APPLY to make consistent copies of table spaces and index spaces," on page 265.

TARGET-JOB-REPAIR-DDN

This field is used to generate REPAIR jobs to correct DB2 catalog inconsistencies that can occur when DB2 objects are copied from one subsystem to another. The inconsistencies processed by DB2 Cloning Tool Table Space Cloning are record format (BRF vs. RRF), actual page format vs. catalog column RBA_FORMAT, HASHDATAPAGES, and versioned objects. Record format, page format, and HASHDATAPAGES processing only apply to DB2 V11 and may only be run on table spaces. Versioned object processing applies to all DB2 versions; DB2 V9 and V10 REPAIR VERSIONS can be run on table spaces and index spaces. The REPAIR jobs

| can be run automatically at the end of the target job or manually after
| target job completion. Specify 1 to 7 characters; in the target job, the
| DDNAME will be suffixed with an I for the input DD and an O for the
| output DD.

AUTO-START-SOURCE-SPACE

Specify whether the source job should optionally start each source DB2 table space and index space after the copy process is complete. Enter Y to start the source table spaces and index spaces in RW mode after the copy is complete. Enter N to leave source table spaces and index spaces stopped after the copy is complete. Enter R to have the source table spaces and index spaces restored to the status they were before executing the source job; however, if a source table space or index space has an initial status of STOP or STOPP, the table space or index space is started in RW.

If you set this parameter to Y or R, and a table space or index space has a status that is not STOP, RW, RO or STOPP, a RC of 8 is issued and the table space or index space is not copied.

If FUZZY-COPY YES is specified, AUTO-START-SOURCE-SPACE is ignored.

AUTO-START-TARGET-SPACE

Enter YES to allow the target job to optionally start each target DB2 table space and index space after the cloning process is complete. The target spaces are started after the target job completes.

AUTO-STOP-TARGET-SPACE

Enter YES to allow the source job to optionally stop the target DB2 table spaces and index spaces. If the target table spaces and index spaces are already stopped, set this to N. This will prevent the DB2 display command from being issued against all the target table spaces and index spaces.

CHECK-INDEX-KEYS

| Enter YES to specify that additional checks are to be made on index
| compatibility. LIMITKEY is always checked and is normally sufficient. In
| some cases, a mismatch is missed unless the keys read and checked also. If
| using index caching, run once with CHECK-INDEX-KEYS(YES), change
| any indexes to correct the mismatches, and then run with
| CHECK-INDEX-KEYS(NO). Use of this command may cause performance
| degradation, especially when there are many indexes to copy or caching is
| being used to access the catalog (caching is not supported for index keys).

COPY-IF-NO-DB2-TARGET-OBJECTS

Enter YES if the source VSAM data sets are to be copied even if the target DB2 table spaces and index spaces do not exist in the target DB2 catalog. Defaults are used for the high level qualifier (DEFVCAT subcommand from the COPY command) and the 5th level qualifier (F0001) when creating the target data set names.

COPY-IJ-TO-NONEXISTENT-TARGET

| Use this field to create data sets on the target subsystem without the need
| to rename and delete temporary data sets on the target. This field can be
| used when the target data sets have not yet been created and if the target
| objects were not created using DEFINE NO. When this field is set to NO
| and no target data sets are found (and the objects were not defined using
| DEFINE NO), the data sets are copied from source to target using F0001 as
| the fifth-node qualifier. When this field is set to YES, you must also specify
| the target data set VCAT using the OBJECT-TRANSLATE VCAT command

| or the TARGET-DB2 DEFVCAT command. If the target data set VCAT is
| not supplied, the copy will not be performed. The target objects are created
| as follows:

- | • If the objects exist in the target catalog, the data sets are copied from the
| source using the I or J fifth-node qualifier that appears in the target
| catalog.
- | • If the objects do not exist in the target catalog, the data sets are copied
| from the source using the source catalog object I or J fifth-node qualifier
| for the target data sets.

DATA-MASKING

Enter YES to enable data masking. This feature enables data to be obscured or transformed during the table space cloning process.

DSNS-PER-COPY

Specify the number of data sets to include in each DSS copy command.

DSS-COPY-COMMANDS

Enter the number of DSS copy commands to be included in each DSS copy command.

EXTEND-TARGET-PBG-TABLESPACE

| Specify YES in this field when a source partition-by-growth table space has
| more partitions than the corresponding target partition-by-growth table
| space and you want to add partitions to the target. As many target
| partitions are added as needed to match the number of source partitions.
| Target partitions will be added even if DATA-MOVER(PGM(NONE)) is
| specified.

DB2 10 OR LATER

| This field is used with the EXTEND-TARGET-PBG-TABLESPACE field, and
| determines how PBG partitions are added on the target. Specify YES for
| DB2 10 and later subsystems; ALTER TABLE will be used to add partitions
| to the target. Specify NO for DB2 9 or earlier subsystems; DB2 UNLOAD
| and LOAD will be used to add partitions. When you build the jobs, the
| UNLOAD and LOAD DDs will be added to the source job (and TCP/IP
| job if required).

FUZZY-COPY

Indicate whether the source table spaces and index spaces should be stopped before replicating them. If you specify YES, either ADRDSSU is invoked with TOLERATE(ENQF) or the EMC API is invoked with TOLERATEENQFAILURE(Y). Note that for ADRDSSU, RACF authority for TOLERATE(ENQF) will be required.

CAUTION:

This procedure may cause data integrity issues.

INCLUDE-ALL-RI

Enter YES to indicate that referentially related table spaces and index spaces are to be included in the list. The RI indicator is automatically inserted in all LISTDEF statements when YES is specified.

LONGVAR-COMPATIBILITY

| Enter YES enable LONGVAR compatibility. When objects are migrated to
| DB2 Version 9.1, all LONGVAR columns remain. However, new
| LONGVAR columns become VARCHAR columns. If running a source job
| where the source objects have LONGVAR column(s) and the target objects
| have corresponding VARCHAR column(s), or vice versa, a mismatch will

| be reported via a warning message. Entering YES in this field prevents the
| mismatch message and the return code 4. Note that the lengths of the
| corresponding columns must be the same. If not, data may be truncated or
| a DB2 abend may occur.

REPLACE-TARGET-DSN

Specify Y to replace (overwrite) the target VSAM object if it exists. For this parameter, the I and J data sets are considered the same data set. For example, if the data set DSN091D.DSNDBC.DAHDB.DAH2TS.J0001.A001 exists on the target and DSN091D.DSNDBC.DAHDB.DAH2TS.I0001.A001 is the target data set name, the copy will not be allowed if N is specified in REPLACE-TARGET-DSN.

When the value is N, another data set will be created with the same name as the target data set except for the 5th level qualifier. Note that the target 5th level qualifier will be obtained from the target catalog for the table space or index space being copied. If there is no target table space or index space and parms allow the source data set to be copied if no DB2 table space or index space exists, F0001 is used.

RESET-LOGRBA

Enter YES to indicate that the target job should reset the LOGRBA in the copied table space and index space data sets. The LOGRBA will always be reset if there are OBID changes to be made. The level IDs in the target VSAM objects are always reset to prevent DB2 down-level rejection of the target VSAM objects. If you specify NO, the DB2 table space or index space may be unusable after completion of the target job.

SIM

Specify if you want to run table space cloning in SIMULATE mode. Enter N to run in normal mode. Enter Y to process LISTDEF to get source table spaces and index spaces and stop. Enter A to specify one of the following:

- If PGM(ADRDSU) or PGM(EMCAPI), stop target and then source spaces, call ADRDSU in NORUN mode, start source and target spaces and write out SYNCDB2 commands for the target.
- If PGM(NONE), validate target table spaces and index spaces and write out SYNCDB2 commands for the target.

CAUTION:

| **DDL processing cannot be simulated. If you do not want DDL to be**
| **processed during a simulation, change DDL-ENABLE to N.**

V7-MIGRATED-OBJECTS-PRESENT

Enter YES if the table space cloning will be copying migrated DB2 V7 objects on a DB2 Version 8 or DB2 Version 9.1 subsystem.

Note: Setting this value to YES will cause performance degradation.

CHECK-DATASET-COMPATIBILITY

| Enter YES to specify that several VSAM attributes should be checked for
| compatibility between source and target subsystems. The following
| attributes are checked:

- If the data set is in extended format.
- If the data set was allocated using extended addressability.
- Whether the data set can be compressed.
- Whether the data set is striped.
- Whether the data set can be spanned.

| These attributes must be the same between the source and target
| subsystems. When one or more data set incompatibilities are found, no
| copies are attempted and the source job ends with RC=8. Use PGM(NONE)
| to check data set compatibility. Correct the data sets found to be
| incompatible. Then use PGM(ADRDUSSU) to make the copies.

IGNORE-RF-MISMATCH-IF-NO-VAR-COLS

| Enter YES to allow a copy of table spaces from source to target when there
| is a mismatch involving reordered row format. For example, the source
| may have been migrated from DB2 V8 and be in basic row format (BRF)
| and the target objects may have been created on DB2 V9 NFM and be in
| reordered row format (RRF). When IGNORE-RF-MISMATCH-IF-NO-VAR-
| COLS is set to YES, table spaces with no variable columns can ignore the
| mismatch in row format and be copied without a warning. Variable
| columns are VARCHAR, LONGVAR, VARG, LONGVARG and VARBIN.

WARN-IF-OBJECT-NOT-TRANSLATED

| Enter YES if the table space cloning job is to check that each source object
| is translated to a new target value. Source objects that are not translated
| result in a warning message. Use this with DATA-MOVER PGM(NONE) to
| check that all masks are set correctly.

WARN-ON-DATASET-EXTENSION-MISMATCH

| Use this field to specify the type of message that is issued when the
| number of data set extensions differs from source to target. If you specify
| YES, a warning message is issued and the job return code is set to 4. If you
| specify NO, an informational message is issued.

WARN-ON-INCOMPLETE-RI

| Enter YES to specify that when one or more LISTDEF statements do not
| specify RI, a warning message is issued.

WARN-ON-SIMPLE-TABLESPACE

| Enter YES to specify that you want to be warned when simple table spaces
| are to be copied. When NO is specified, an informational message is
| printed for each simple table space found on the source or target
| subsystem. When YES is specified, a warning message is issued.

| In some cases when a simple table space is copied, the copy may have
| duplicate rows. This is caused by incompatibilities between table spaces
| migrated from an earlier DB2 version and processed using DB2 Version 9.1
| or later.

| When a simple table space is copied for the first time, ensure the target
| table space is accessible and has the correct number of rows. Once all of
| the simple table spaces in a job are verified, use the default NO to
| eliminate warning messages.

CATALOG-PREFETCH ENABLE-PREFETCH

| Enter YES to enable catalog prefetch. The CATALOG-PREFETCH
| parameter allows all the rows in SYSIBM.SYSTABLESPACE,
| SYSIBM.SYSTABLEPART, SYSIBM.SYSTABLES and SYSIBM.SYSCOLUMNS
| for table spaces and SYSINDEXES and SYSINDEXPART for indexes to be
| read without a WHERE clause. You must also enable source or target
| prefetch as desired using the ENABLE fields that follow. To enter specific
| databases to be included in the CATALOG-PREFETCH parameter, enter S
| in the Option line for source catalog prefetch or T in the Option line for
| target catalog prefetch and press Enter.

ENABLE-SOURCE-PREFETCH

Enter YES to enable prefetch for the source catalog. When enabled, source objects from one or more databases will be saved in memory (cached) during a single pass of the catalog tables. If this parameter is set to YES and the SOURCE-PREFETCH-DATABASE-LIST exists, it will be used. If SOURCE-PREFETCH-DATABASE-LIST does not exist, LISTDEF will be used to find objects to prefetch. To specify databases to be prefetched for the source catalog, enter S in the Option line and press Enter.

ENABLE-TARGET-PREFETCH

Enter YES to enable prefetch for the target catalog. When enabled, the target objects from one or more databases will be saved in memory (cached) during a single pass of the catalog tables and the target objects will be mapped from the source objects. This includes object translation if required. When target prefetching is enabled and the TCP/IP server job is in use, target objects and the enable flag come from the source job. To specify databases to be prefetched for the target catalog, enter T in the Option line and press Enter.

INCLUDE OBJECT-TRANSLATE

Enter YES if you plan to use the OBJECT-TRANSLATE parameter to map the source objects to target objects with different names. This feature allows you to copy table spaces and index spaces to the same subsystem. To specify values for the OBJECT-TRANSLATE parameter, enter O in the Option line and press Enter.

INCLUDE JOB-TEMPLATE

Enter YES if you plan to use job templates. To specify job template data set and DD names, enter J in the Option line and press Enter.

Specifying source and target CATALOG-PREFETCH databases:

When you enter YES for the CATALOG-PREFETCH ENABLE-PREFETCH field on the DB2 tablespace clone COPY Command panel, you may also specify list of databases to be prefetched when accessing the source and/or target catalog.

Enter S in the **Option** field on the DB2 tablespace clone COPY Command panel to access the Specify Source Prefetch Databases panel. Enter T in the **Option** field on the DB2 tablespace clone COPY Command panel to access the Specify Target Prefetch Databases panel.

The panels are identical except for specifying the source or the target databases.

Specify the names of the databases to be prefetched on the appropriate panel as follows:

Source Database

Specify a list of databases to be prefetched when accessing the source catalog. When using LISTDEF with more than one database, multiple reads of the same catalog tables can take place. Multiple reads can be eliminated by using this prefetched list. Duplicate databases are not detected and cause no problems. More than 1,000 databases can be entered.

Important: If all databases are not included in the prefetch database list, the objects will not be found and the source job will not run correctly.

Target Database

Specify a list of databases to be prefetched when accessing the target catalog. Connection to the target can be CAF, DDF or TCP/IP. This

subcommand can be used in the source job and is passed to the TCP/IP server job if it is in use. More than 1,000 databases can be entered. If this command is not entered, the list of target databases to be cached is generated from the source data set mapped to target names using object translate. When copying a large number of table spaces, compare source job run times with and without this database list to determine which gives the best performance.

Important: If all databases are not included in the prefetch database list, the objects will not be found and the source job will not run correctly.

You can add an input line to include another database in the prefetch list by entering A in the **Command** field. To delete a database from the prefetch list, enter D in the line command area next to the database.

Specifying OBJECT-TRANSLATE defaults:

When you enter YES for the INCLUDE OBJECT-TRANSLATE field on the DB2 tablespace clone COPY Command panel, you need to specify the names and object types of the source and target objects. The OBJECT-TRANSLATE parameter renames the target table spaces and index spaces with the supplied names. This feature allows you to copy table spaces and index spaces to the same subsystem or to a different target subsystem with different names.

Enter O in the **Option** field on the DB2 tablespace clone COPY Command panel to access the DB2 tablespace OBJECT-TRANSLATE Command panel.

Specify the names of the source objects and the new target object names, in pairs.

OBJTYPE

Enter the object type that will be copied in this field. Specify one of the following:

- CR - Object creator
- DB - Database
- TS - Table space
- IX - Index
- IS - Index space
- TB - Table
- VC - VCAT

SOURCE NAME/TARGET NAME

Specify the names of the source objects and the new target object names, in pairs. The source and object pairs may be specified using masking, as follows:

- Percent sign (%) or asterisk (*) represents n characters.
- Underscore (_) or question mark (?) represents a single character. Use the question mark (?) rather than the underscore (_) for creator, table and index names, as the underscore is a valid character for these three object names

You can add an input line to include another pair of source and target spaces by entering A in the **Command** field. To delete a pair of source and target spaces, enter D in the line command area next to the pair.

Specifying job template data set and member name defaults:

When you specify YES for the INCLUDE JOB-TEMPLATE field on DB2 tablespace clone COPY Command panel, the job statements are generated using template input DDs and output DDs specified in the JOB-TEMPLATE subcommand. Job templates consist of the z/OS JCL statements, DSS commands, user variables, and processing variables that DB2 Cloning Tool Table Space Cloning uses for input. You need to supply DD information for the templates.

Enter J in the **Option** field on the DB2 tablespace clone COPY Command panel to access the Specify Job Template Data Sets and Members panel. This panel allows you to define the default data sets and DD names for building jobs using job templates.

TEMPLATE INPUT DATA SET

Enter the data set name that contains the job template members.

TEMPLATE OUTPUT DATA SET

Enter the data set name that will contain the output built using the job templates after the source job is executed.

INPUT DD DISP

Enter the data set disposition of the template input data set.

OUTPUT DD DISP

Enter the data set disposition of the template output data set.

INPUT MEMBER

Enter a member name that contains a job template. The member must exist in the data set defined in the Template Input Data Set field. Detailed information about setting up job templates can be found in the sample members CKZJOB1, CKZJOB2, and CKZJOB3 provided in the product sample library.

INPUT DDNAME

Enter a DD name to be used to point to the job template input data set.

OUTPUT MEMBER

Enter a member name that will contain the job built from the job templates after the source job has been executed.

OUTPUT DDNAME

Enter a DD name to be used to point to the job template output data set.

You can add an input line to include another pair of input and output template members by entering A in the **Command** field. To delete a pair of input/output members and DDs, enter D in the line command area next to the pair.

Specifying DDL attribute change defaults:

When you specify YES for the PROCESS-DDL field on the DB2 tablespace clone COPY Command panel, DB2 Cloning Tool Table Space Cloning generates and/or executes DDL to be used for creating non-existent target objects. The DDL can also change values from source to target that are not changed with object translate, such as LOG, PRIQTY and SECQTY, or CLOSE.

Enter D in the **Option** field on the DB2 tablespace clone COPY Command panel to access the DB2 tablespace DDL-ATTRIBUTE-CHANGE Command panel. This panel allows you to define the values to be changed and specify to which objects you want the changes to apply.

To create an attribute change definition, enter C in the Command line and press Enter. The Edit DDL-ATTRIBUTE-CHANGE Command is displayed.

Attribute Name, Source Value, Target Value, Apply to Type

For these fields, refer to the following table. This table lists the possible Attribute Name parameters, their possible values (Source Value and Target Value), and the object that the attribute change may apply to (Apply to Type).

Table 46. Possible attribute change values

Attribute Name	Possible source/target values	Apply to Type
STOGRUP	For DB2 Version 8 ENFM, Version 9.1 and later: 128 chars For DB2 Version 8 CM: 8 chars	Database, table space partition, index partition
GBPCACHE	CHANGED ALL SYSTEM NONE	Table space partition
	CHANGED ALL NONE	Index partition
LOG	YES NO	Table space
	Note: LOG YES or NO syntax is generated for DB2 Version 8 LOGGED YES or NO syntax is generated for DB2 Version 9.1 and later	
PRIQTY	1-7 decimal digits or -1	Table space partition, index partition
SECQTY	1-7 decimal digits or -1	Table space partition, index partition
	Note: PRIQTY and SECQTY are matched using the catalog value of PQTY and SECQTYI, not what is entered using the DDL. DDL generation multiplies this catalog value by 4 to get the PRIQTY and SECQTY DDL values. For user managed data sets, RUNSTATS must have been run to populate the fields in the catalog.	
TRACKMOD	YES NO	Table space partition
CLOSE	YES NO	Table space, index
DATAcapture	NONE CHANGES	Table
BUFFERPOOL	<i>pool name</i>	Database, table space, index

Apply To Object

This field specifies which source object names this change applies to. To specify all objects, leave this field blank or enter a percent sign (%).

You can also specify masking, but the mask must refer to a database, table space, table, or index. When using partitioned table spaces or partitioned index spaces in the Apply to Type field, use the table space or index space name in the Apply To Object field.

% represents zero or more characters. ? represents one character. An underscore is NOT considered a single masking character for DDL attribute changes.

| When you have specified all DDL attribute changes, press PF3.

| Once you have created an attribute definition, you can enter E in the line
| command area to edit it. Enter V to view the definition without changing it. Enter
| R to repeat (copy) a definition and edit it. Enter D to delete the definition.

| **Specifying LOG-APPLY defaults:**

| The LOG-APPLY keyword allows DB2 log records written before the copies in the
| source job are made until the target job is run to be applied to the target objects.
| This feature eliminates the need to stop and start source objects to achieve a
| consistent copy.

| Enter L in the **Option** field on the DB2 tablespace clone COPY Command panel to
| access the DB2 tablespace LOG-APPLY Command panel. This panel allows you to
| define defaults for the LOG-APPLY keyword.

| The fields on this panel are:

| **LA-ENABLE**

| Enter YES to enable log apply. If set to NO, the parameters are validated,
| but no logs are added to the copied data set.

| When cloning table spaces from image copies, you must specify YES in this
| field. In addition, you must ensure that the source subsystem SDSNEXIT
| library is included in the STEPLIB of the source job. For normal table space
| cloning jobs that use the LOG-APPLY feature, you must ensure that the
| source subsystem DB2 SDSNEXIT library is in the STEPLIB for both the
| source job and the target job.

| **QUIESCE-POINT**

| Enter YES to specify that a quiesce point is to be issued after the copies are
| complete.

| **WARN-IF-SKIP-QUIESCE**

| Enter YES to issue a warning message if QUIESCE must be skipped for a
| status that prevents a quiesce point.

| **COMMON-CONSISTENT-POINT**

| Enter YES to specify that all table space objects are to be brought to the
| same common consistent point. If no common consistent point can be
| found, the spaces are not processed (unless YES is entered in the
| UNIFIED-WARNING field).

| **UNIFIED-WARNING**

| This field is only valid if YES was specified in the COMMON-
| CONSISTENT-POINT field. If a common consistent point cannot be found
| for the included spaces, but you want processing to continue with a
| warning, enter YES in this field. If no common consistent point is found
| and NO was entered in this field, processing is halted and error messages
| produced.

| **NUMBER-OF-BUFFERS**

| This field is a QSAM override for the BUFNO= parameter in JCL; use this
| field to adjust the number of buffers if desired.

| **NUMBER-OF-CHANNEL-PROGRAM**

| This field is a QSAM override for the NCP= parameter in JCL; use this
| field to adjust the number of read or write macro instructions if desired.

SORT-PROGRAM

Specify the sort program to be used; enter DFSORT or SYNCSORT.

USE-TCPIP

Specify Y in this field to enable log apply processing when you are cloning table spaces from one LPAR to a different LPAR. To enable cross-LPAR log apply processing, you must also set up and submit the source TCP/IP server job. Specify N in this field when the source and target table spaces are on the same LPAR.

REBUILD-COPY-NO-INDEXES

This field is used only when cloning table spaces from image copies. Enter YES to include jobs to rebuild indexes that were created or altered with COPY NO as part of the target job.

END-POINT TYPE

This field is used only when cloning table spaces from image copies. Indicate the end point at which the log apply process is to stop processing logs:

- **CURRENT**: Processes up to the current point in time.
- **QUIESCE**: Processes to the last quiesce point.
- **LOGPOINT**: Process to a specific point in the log. If you specify this value, you must also specify a log point in the TO-LOGPOINT field.

TO-LOGPOINT

This field is used only when cloning table spaces from image copies. If you specified LOGPOINT in the END-POINT-TYPE field, enter the log point (RBA or LRSN) in 12 or 20 hexadecimal digits. If DSNJCNVN is set and 10 byte RBAs are in use, all byte strings must be 10 bytes. When this value is specified, logs are applied up to this specific log point.

WARN-IF-TS-DEFINED-NOT-LOGGED

This field is used only when cloning table spaces from image copies. When a base or LOB table space has the NOT LOGGED attribute, DB2 does not create logs for the space. This could result in errors when the source object is copied to the target. If WARN-IF-TS-DEFINED-NOT-LOGGED is set to Y, a warning message is output for each table space with the NOT LOGGED attribute. If WARN-IF-TS-DEFINED-NOT-LOGGED is set to N, an informational message is output for each table space with the NOT LOGGED attribute.

There are several data sets used during LOG-APPLY processing that you must set options for.

- Enter M in the **Command** line to set data set options for the minilog data set.
- Enter S in the **Command** line to set data set options for the sort file data set.
- Enter W in the **Command** line to set data set options for the work file data set.

Refer to the following topics for more information about the default values for these data sets.

Minilog data set options

The minilog data set is required for LOG-APPLY processing.

MINILOG HLQ

Enter the high level qualifier for the minilog data set.

| **SPACES-PER-MINILOG**

| Enter the maximum number of spaces that a minilog will track.

| **MINILOG-LARGE-FILE-TYPE**

| Enter YES to specify that dynamic allocation of the minilog data set should
| include the LARGE attribute. This allows for data sets to exceed 65,535
| tracks.

| **MINILOG-UNIT-TYPE**

| Enter the unit type for the minilog data set.

| **MINILOG-QUANTITY-IN-TRACKS**

| Enter YES if the minilog is to be allocated in tracks. If you enter NO, the
| minilog is allocated in cylinders.

| **MINILOG-PRIMARY-QUANTITY**

| Enter the minilog data set's primary quantity.

| **MINILOG-SECONDARY-QUANTITY**

| Enter the minilog data set's secondary quantity.

| **MINILOG-VOLUME-COUNT**

| Enter the maximum number of volumes that the minilog data set will
| require. If SMS parameters are specified to allow for multi-volume DASD
| data sets, this parameter must be at least as large as the number of
| volumes that the minilog will ultimately occupy. Enter a value between
| 1-255 inclusive, or blank to omit the volume count parameter.

| **MINILOG-DATACLAS**

| If the minilog data set will be managed by SMS, enter the SMS Data Class
| here.

| **MINILOG-STORCLAS**

| If the minilog data set will be managed by SMS, enter the SMS Storage
| Class here.

| **MINILOG-MGMTCLAS**

| If the minilog data set will be managed by SMS, enter the SMS
| Management Class here.

| **Sort file options**

| The Set SORTFILE options panel allows you to enter default settings for the sort
| file to be used by DFSORT or SYNCSORT.

| **SORTFILE-LARGE-FILE-TYPE**

| Enter YES to specify that dynamic allocation of the sort file data set should
| include the LARGE attribute. This allows for data sets to exceed 65,535
| tracks.

| **SORTFILE-UNIT-TYPE**

| Enter the unit type for the sort file data set.

| **SORTFILE-QUANTITY-IN-TRACKS**

| Enter YES if the sort file is to be allocated in tracks. If you enter NO, the
| sort file is allocated in cylinders.

| **SORTFILE-PRIMARY-QUANTITY**

| Enter the sort file data set's primary quantity.

| **SORTFILE-SECONDARY-QUANTITY**

| Enter the sort file data set's secondary quantity.

|
| **SORTFILE-VOLUME-COUNT**

| Enter the maximum number of volumes that the sort file data set will
| require. If SMS parameters are specified to allow for multi-volume DASD
| data sets, this parameter must be at least as large as the number of
| volumes that the sort file will ultimately occupy. Enter a value between
| 1-255 inclusive, or blank to omit the volume count parameter.

| **SORTFILE-DATACLAS**

| If the sort file data set will be managed by SMS, enter the SMS Data Class
| here.

| **SORTFILE-STORCLAS**

| If the sort file data set will be managed by SMS, enter the SMS Storage
| Class here.

| **SORTFILE-MGMTCLAS**

| If the sort file data set will be managed by SMS, enter the SMS
| Management Class here.

| **Work file options**

| The Set WORKFILE options panel allows you to enter default settings for the work
| file to be used by log apply. Several work files are required for log apply. These
| values should be set to large default values.

| **WORKFILE-LARGE-FILE-TYPE**

| Enter YES to specify that dynamic allocation of the work file data set
| should include the LARGE attribute. This allows for data sets to exceed
| 65,535 tracks.

| **WORKFILE-UNIT-TYPE**

| Enter the unit type for the work file data set.

| **WORKFILE-QUANTITY-IN-TRACKS**

| Enter YES if the work file data set is to be allocated in tracks. If you enter
| NO, the work file data set is allocated in cylinders.

| **WORKFILE-PRIMARY-QUANTITY**

| Enter the work file data set's primary quantity.

| **WORKFILE-SECONDARY-QUANTITY**

| Enter the work file data set's secondary quantity.

| **WORKFILE-VOLUME-COUNT**

| Enter the maximum number of volumes that the work file data set will
| require. If SMS parameters are specified to allow for multi-volume DASD
| data sets, this parameter must be at least as large as the number of
| volumes that the work file data set will ultimately occupy. Enter a value
| between 1-255 inclusive, or blank to omit the volume count parameter.

| **WORKFILE-DATACLAS**

| If the work file data set will be managed by SMS, enter the SMS Data
| Class here.

| **WORKFILE-STORCLAS**

| If the work file data set will be managed by SMS, enter the SMS Storage
| Class here.

| **WORKFILE-MGMTCLAS**

| If the work file data set will be managed by SMS, enter the SMS
| Management Class here.

Viewing ZPARM defaults

To view the ZPARM settings for the subsystem specified in the LOCAL-SSID field on the DB2 tablespace clone SET command panel, enter Z in the **Option** field on the DB2 tablespace LOG-APPLY Command panel. The View ZPARM settings panel displays the following read-only fields:

ZPARM-MEMBER

This field displays the default ZPARM member name for the LOCAL-SSID that was specified on the DB2 tablespace clone SET command panel.

DATA-SHARING-MEMBERS: SSID, ZPARM, BSDS01 DSN, BSDS02 DSN

If the LOCAL-SSID is a data sharing group, these fields display information about the data sharing member's ZPARMs and BSDSs. These fields are blank if the LOCAL-SSID is a non-data sharing group.

When you have finished specifying LOG-APPLY options, press PF3 to return to the DB2 tablespace clone COPY Command panel.

Specifying OBJECT-MISMATCH-RETURN-CODE defaults:

In most cases, attributes of the source and target table spaces and index spaces must match for a successful cloning. However, you can override the default return codes for specific object mismatches.

About this task

If a return code is not specified for a mismatch, the default return code of 4 is used. Object mismatches are detected after the source and target objects are identified, and before any copies are started.

Procedure

1. Enter A in the **Command** line and press Enter. The Select Object Mismatch Type panel is displayed.
2. Use the S line command to select one or more mismatch types from the displayed list of possible mismatch types.
3. When you have finished selecting all the mismatch types, press Enter. The DB2 tablespace OBJECT-MISMATCH-RETURN-CODE Command panel is displayed.
4. The return code for each selected mismatch type defaults to 4. To specify a different return code for a mismatch type, update the code in the RC column next to the mismatch type.

HLQDDDF command defaults

The HLQDDDF command is used in table space cloning to pass input and output DDs to ADRDSSU.

The following values can be set on the table space cloning HLQDDDF command defaults panel. These values are used strictly as defaults for HLQDDDF command parameters when you are creating a new table space cloning profile. The parameter values can later be edited in the profile.

HLQNAME

Specify the high level qualifier this command applies to.

DIR Specify whether this DD name or names are provided to ADRDSSU as an input or output DD. Enter IN for input and OUT for output.

DD Names

Enter the DD names to pass to ADRDSSU as an input or output DD.

You can add an input line to include another high level qualifier and DD names by entering A in the Command line. To delete a line, enter D in the line command area next to the HLQNAME.

Setting the DD parameters

The first time you enter a high level qualifier and its associated DD names, you must enter the E line command next to the HLQNAME to enter data definition parameters for its DDs. You can also edit the data definitions at any time using the E line command.

Specifying HLQDDDF DD data definitions:

When you enter E next to a DD on the DB2 tablespace clone HLQDDDF Command panel, the HLQDDDF DD Specification panel is displayed. This panel allows you to specify the data definitions for the HLQDDDF DDs.

DD Name

If you edited an existing high level qualifier and DD name line, the DD name selected on the previous panel appears in this field. You can enter additional DD names in the space provided if desired.

DD Enter data definition parameters for the DD name in this line.

Defining an XML object

If you are planning to copy DB2 tables that contain XML columns, you must create a table with an XML column that DB2 Cloning Tool Table Space Cloning will use for adding strings to SYSIBM.SYSXMLSTRINGS.

Procedure

1. Enter 5 on the User DB2 tablespace clone settings menu. A window is displayed with data set name and member fields.
2. Enter the data set name and member to hold the DDL to create the tables. The data set and member are derived from the CKZCRXML DD name on the DB2 tablespace clone DD specification panel. This DD must be entered and have its data definition defined on that panel before you can continue. If the data set does not exist, it will be created for you.
3. Press Enter. An ISPF edit session is provided for you to enter the DDL. For a sample of the DDL to create the XML table, refer to the CKZXMDDL member in the SCKZJCL sample library.

Creating cloning jobs using the interface

This topic describes some basic procedures for subsystem cloning and table space cloning using the ISPF interface.

Before you begin

Before you begin creating subsystem or table space cloning jobs using the ISPF interface:

- You must ensure that the subsystems that you want to use as source and target subsystems have been added and configured using the Administrator functions portion of the interface.

- You should verify that the user configuration settings are appropriate for your needs.

Basic procedure

The basic procedure to create cloning jobs for both subsystem and table space cloning is:

1. Create a cloning profile.
2. Select the menu options to specify the type of cloning, the DDs, command settings, and other options.
3. Build the profile. Building the profile creates the series of jobs that will perform the cloning.
4. Review the generated jobs to ensure the output is as desired.
5. To invoke the cloning process, submit the generated jobs. The member names are generated alphanumerically; simply submit the jobs in order.

Creating a profile

To create a new subsystem or table space cloning profile, follow these steps.

About this task

The process of creating a profile is the same for subsystem cloning and table space cloning.

Procedure

1. On the Primary Option menu, enter option 1.
2. On the Clone menu, enter option 1 for subsystem cloning or 2 for table space cloning.
3. On the Enter Clone Profile Selection Criteria window, enter selection criteria (if desired). Standard ISPF wildcarding is allowed.
4. On the DB2 Subsystem Clone Profile Display or the DB2 Tablespace Clone Profile Display, enter C in the Command field. The Enter New DB2 Subsystem Clone Profile Options or Enter New DB2 Tablespace Clone Profile Options window is displayed.
5. Enter a profile name and share option and press Enter. The Edit DB2 Subsystem Clone Profile menu or the Edit DB2 Tablespace Clone Profile menu is displayed.
6. To set up the cloning profile, select the options on the menu.

Results

Once a profile has been created, it can be edited, renamed, viewed, copied or deleted.

Subsystem cloning

This section describes the basic steps for subsystem cloning using the ISPF interface.

Start the ISPF interface and create a subsystem cloning profile. After the cloning profile has been created, the Edit DB2 Subsystem Clone Profile menu is displayed, as shown in the following figure:

```

                                Edit DB2 Subsystem Clone Profile
Option ==>

Creator . . . : TWUSR           Name . . . . : TEST2
Share Option . : UPDATE        Description . .

1  Select Source and Target DB2 subsystems
2  Select Source and Target Volume Pairing
3  Select Source and Target ICF catalogs
4  Select Rename masks
5  Select other parameters

```

Subsystem cloning steps summary

The general steps for using the ISPF interface to clone a DB2 subsystem are described in this topic.

Procedure

1. Ensure the source and target subsystems have been added and properly configured under the Administrative Options option.
2. Create a subsystem cloning profile.
3. Select the source and target subsystems and specify the cloning type (option 1 on the Edit DB2 Subsystem Clone Profile panel).
4. Verify or enter source and target HLQs (option 1 on the Edit DB2 Cloning Profiles panel).
5. Specify source and target stogroup or volume pairings (option 2 on the Edit DB2 Subsystem Clone Profile panel).
6. Specify source and target ICF catalogs (option 3 on the Edit DB2 Subsystem Clone Profile panel).
7. Specify rename masks (option 4 on the Edit DB2 Subsystem Clone Profile panel).
8. Build the profile.
9. Submit the jobs (in order).

Step summary for subsystem cloning from a system-level backup

The general steps for using the ISPF interface to clone a DB2 subsystem from a system level backup are described in this topic.

Procedure

1. Ensure the source and target subsystems have been added and properly configured under the Administrative Options option.
2. Create a subsystem cloning profile.
3. On the Edit DB2 Subsystem Clone Profile panel, enter option 1 to specify the source and target subsystems and specify cloning type of online.
4. On the Edit DB2 Subsystem Clone Profile panel, enter option 2 to specify source and target volume pairing.
5. On the Select Source and Target Volume Pairing panel, enter option 3 to select Source System Level Backup.

6. On the Set Source System Level Backup panel, enter DB2SLB in the **System Level Backup type** field.
7. To specify source and target volume pairing, on the Set Source System Level Backup panel, do one of the following:
 - To use storage group names, enter YES in the **Pair using Source Storage Group names** field. You must also specify the source and target storage groups by using options 1 and 4 on the Select Source and Target Volume Pairing panel.
 - To use volume serials, enter NO in the in the **Pair using Source Storage Group names** field. You must also specify either target volumes or target storage groups by using options 4 or 5 on the Select Source and Target Volume Pairing panel.
8. If you want to use SLB dump tapes as the source for the cloning:
 - a. Enter YES in the **Use Dumptapes** field.
 - b. Enter R in the Command field.
 - c. On the RESTORE-FROM-DUMPTAPES command panel, verify the settings for the RESTORE-FROM-DUMPTAPES command.
9. Build the profile.
10. Submit the jobs (in order).

Select source and target subsystems

You must first select the source and target subsystems and specify offline or online cloning.

For non-data sharing source subsystems

Select the source and target subsystems and specify offline or online cloning.

Procedure

1. On the Edit DB2 Subsystem Clone Profile menu, enter option 1.
2. On the Select Source and Target DB2 Subsystems menu, enter A in the Command field.
3. On the Select Source DB2 Subsystem panel, select the source subsystem and press Enter.
4. On the Select Cloning Type panel, enter ONLINE or OFFLINE in the **Type of cloning** field.
5. Press Enter. The Select Target DB2 Subsystem panel is displayed.
6. Select a target subsystem and press Enter. The Edit DB2 cloning values panel is displayed.

For data sharing source subsystems

Select the source and target subsystems and specify offline or online cloning. In addition, for source subsystems that are data sharing, you must specify the data sharing attributes of the target subsystem.

Procedure

1. On the Edit DB2 Subsystem Clone Profile menu, enter option 1.
2. On the Select Source and Target DB2 Subsystems menu, enter A in the Command field.
3. On the Select Source DB2 Subsystem panel, select the source subsystem and press Enter.

4. On the Select Cloning Type panel, enter ONLINE or OFFLINE in the **Type of cloning** field.
5. Enter SAME, FEWER, or NONDS in the **Data sharing attributes of target** field. Specify SAME when the target will be a data sharing group with the same number of members as the source. Specify FEWER if the target will be a data sharing group with fewer members than the source. Enter NONDS when the target will not be a data sharing group.
6. The Select source members to clone panel is displayed. This panel lists all data sharing group members in the source data sharing group. The data sharing group member that you selected as a source subsystem on the Select Source DB2 Subsystem panel is automatically selected. You can select one or more additional data sharing group members to clone.
7. When you have selected all the members to be cloned, press Enter.

Specifying the target subsystem for data sharing source members:

You must specify which members of the target data sharing group are to be used as targets.

If the target data sharing group will have the SAME number of members

1. The Select Target DB2 member panel is displayed. On this panel, select the target subsystems to be paired with source subsystems. The target subsystem you select will be paired with the source member listed near the top of the panel.
2. Select a target by entering S next to the target. When you press Enter, if there are more source subsystems that need to be paired with targets, the panel will be displayed again.
3. Continue to pair target subsystem with source subsystems until all source and targets have been paired; when you press Enter, the Edit DB2 Cloning values panel is displayed.

If the target data sharing group will have FEWER members

If the target is a data sharing group that will have fewer members than when built, then you must specify which target members will be the “surviving members”. For example, if you plan to clone one subsystem to a target data sharing group that has two members, then you will need to specify which target group member will be the surviving member.

1. The Select Target DB2 member panel is displayed. On this panel, select the target subsystems to be paired with source subsystems. The target subsystem you select will be paired with the source member listed near the top of the panel.
2. Select a target by entering S next to the target. When you press Enter, if there are more source subsystems that need to be paired with targets, the panel will be displayed again. Continue to pair target subsystem with source subsystems until all source and targets have been paired.
3. The Manage surviving target DB2 members panel is displayed. On this panel, you can add or delete the surviving member(s) for the target data sharing subsystem. Add a surviving target member by entering A in the Command line and press Enter.
4. The Select surviving target DB2 members panel is displayed. Select the surviving member by entering S in the line command area next to the target subsystem. Or select all listed members by entering A in the command line. Press Enter.

5. The Manage surviving target DB2 members panel is displayed. Each selected surviving member is listed on the panel. The **PRIMARY** and **DDF** fields default to **SOURCE**, which means the target subsystems retain the BSDS attributes of the subsystems. If required, you can edit those fields on this panel.
6. To save changes, press Enter, then PF3. The Enter DB2 cloning values panel is displayed.

If the target subsystem will not be a data sharing group (NONDS)

1. The Select Target DB2 member panel is displayed. On this panel, select the target subsystems to be paired with source subsystems. The target subsystem you select will be paired with the source member listed near the top of the panel.
2. Select a target by entering S next to the target. When you press Enter, if there are more source subsystems that need to be paired with targets, the panel will be displayed again. Continue to pair target subsystem with source subsystems until all source and targets have been paired.
3. When you press Enter, the Select surviving target DB2 member panel is displayed. This panel lists the target subsystem you selected. Since the target will be non-data sharing, this is the only subsystem that will be a surviving member. Enter S in the line command area to select the surviving member.
4. When you press Enter, the Edit DB2 Cloning values panel is displayed.

Add or verify high level qualifiers

The high level qualifiers of the DB2 source and target subsystems must be provided. Verify that they have already been defined when the subsystems were configured under administrative options, or provide the qualifiers using the Enter DB2 HLQs panel.

Procedure

1. On the Edit DB2 cloning values menu, enter option 1.
2. On the Enter DB2 HLQs panel, verify or enter the DB2 HLQs for the source and target subsystem(s).

Specify source and target volume pairings

Specify the input volumes to be copied and the target volumes to which they will be copied.

Procedure

1. On the Edit DB2 Subsystem Clone Profile menu, enter option 2.
2. On the Select Source and Target Volume Pairing panel, enter one of the following:
 - 1 to specify a source SMS storage group or mask that contains the input volumes to be copied.
 - 2 to specify input volumes using volsers or volser masks.
 - 3 to specify a system level backup (SLB) as input for the source
 - 4 to specify one or more SMS storage groups or masks as targets that will be paired with input volumes.
 - 5 to specify the target volumes via volsers or volser masks that will be paired with input volumes.
 - 6 to specify source volumes that are to be excluded from cloning.
 - 7 to specify target volumes that are to be excluded from cloning.

Specify source and target ICF catalogs

Specify the source catalogs that data sets from source (from) volumes are cataloged in, and the corresponding target catalogs that renamed volume data sets are to be cataloged in.

About this task

For each renamed data set, source and target catalog pairs are searched for the ICF catalog the source volume data set was cataloged in. The renamed data set is cataloged in the corresponding target catalog.

Procedure

1. On the Edit DB2 Subsystem Clone Profile menu, enter option 3.
2. On the Select Source and Target ICF catalogs panel, specify the source catalogs that data sets from the source volume are cataloged in, and the corresponding target catalog that renamed data sets on the target volume are to be cataloged in.

Specify rename masks for source and target data sets

The data sets from the COPY step can be renamed onto the target volumes.

About this task

For each renamed data set, source and target catalog pairs are searched for the ICF catalog the source volume data set was cataloged in. The renamed data set is cataloged in the corresponding target catalog.

Procedure

1. On the Edit DB2 Subsystem Clone Profile menu, enter option 4 (Select rename masks).
2. On the Select rename masks panel, enter one of the following:
 - 1 to specify source and target rename masks. On the Rename Masks panel, specify the source and target data set rename masks. The RENAME uses these masks to rename and catalog the data sets from the COPY step onto the target volumes.
 - 2 to exclude data sets from the rename process. The Exclude Masks panel allows you to specify a list of source data set names or masks that will NOT be renamed.

Build the cloning jobs from a profile

Once the profile has been created, build the profile to produce the cloning jobs.

About this task

Begin building the profile on the DB2 Subsystem Clone Profile Display, shown in the following figure:

```

DB2 Subsystem Clone Profile Display
Command ==>                               Scroll ==> CSR

Commands:      C - Create
Line Commands: B - Build D - Delete E - Edit R - Rename V - View C - Copy

Profile Like . . . *
Creator Like . . . TWUSR*

Row 1 of 2
Cmd Name      Creator  Share Option Description
TEST          TWUSR   NO ACCESS
TEST DATA SHARING TWUSRA  UPDATE      TEST DATA SHARING #1

```

Procedure

1. On the DB2 Subsystem Clone Profile Display, enter B next to the profile that you want to build.
2. On the Build DB2 Subsystem Clone jobs panel, enter the data set into which the jobs are to be placed, and specify processing options.
3. Press Enter. The cloning jobs are generated into the specified data set.
4. If you selected the processing option to edit the JCL data set, a panel is displayed listing the jobs that have been generated, as shown in the following figure. You can edit or view the jobs using line commands.

```

Edit TWUSR.CKZ.JCLLIB4
Command ==>                               Scroll ==> CSR

Line Commands: E - Edit V - View

Row 1 of 22
Cmd Name      Created      Changed      ID
ST01C7A1     2009/08/25  2009/08/25  08:59:50 TWUSR
ST01D8A      2009/08/25  2009/08/25  08:59:53 TWUSR
ST02         2009/08/25  2009/08/25  08:59:52 TWUSR
ST03C7A1     2009/08/25  2009/08/25  08:59:50 TWUSR
ST03D8A      2009/08/25  2009/08/25  08:59:53 TWUSR
ST04         2009/08/25  2009/08/25  08:59:52 TWUSR
ST05         2009/08/25  2009/08/25  08:59:53 TWUSR
ST06D9A3     2009/08/25  2009/08/25  08:59:50 TWUSR
ST06T12      2009/08/25  2009/08/25  08:59:53 TWUSR
ST08D9A3     2009/08/25  2009/08/25  08:59:51 TWUSR
ST09D9A3     2009/08/25  2009/08/25  08:59:51 TWUSR
ST10D9A3     2009/08/25  2009/08/25  08:59:51 TWUSR
ST10T12      2009/08/25  2009/08/25  08:59:53 TWUSR
ST11D9A3     2009/08/25  2009/08/25  08:59:51 TWUSR
ST11T12      2009/08/25  2009/08/25  08:59:54 TWUSR
ST12D9A3     2009/08/25  2009/08/25  08:59:52 TWUSR
ST12T12      2009/08/25  2009/08/25  08:59:54 TWUSR
ST13D9A3     2009/08/25  2009/08/25  08:59:52 TWUSR
ST13T12      2009/08/25  2009/08/25  08:59:55 TWUSR
ST14D9A3     2009/08/25  2009/08/25  08:59:52 TWUSR
ST14T12      2009/08/25  2009/08/25  08:59:55 TWUSR
ST22         2009/08/25  2009/08/25  08:59:55 TWUSR

```

Subsystem cloning job reference

This topic provides a list of the cloning job member names and the steps that they perform when executed.

Online cloning jobs

This table provides a list of the member names of the online cloning jobs that are generated and the commands that are run in the jobs.

Table 47. Online cloning job and member reference

Member name	Command name
ST01xxxx	DB2SETLOG SUSPEND
ST02	COPY
ST03xxxx	DB2SETLOG RESUME
ST04	COPYCHECK
ST05	RENAME
ST06xxxx	DB2UPDATE
ST07xxxx	DB2UPDATE (for secondary members of a data sharing group)
ST08xxxx	DB2START
ST09xxxx	DB2START (for secondary members of a data sharing group)
ST10xxxx	DB2FIX (DB2)
ST11xxxx	DB2STOP
ST12xxxx	DB2UPDATE
ST13xxxx	DB2START
ST14xxxx	DB2SQL
ST15xxxx	DB2FIX (application)
ST16xxxx	DB2STOP
ST17xxxx	DB2RBLDBSDS
ST18xxxx	DB2LGRNXCLEAN
ST19xxxx	DB2XCFCLEAN
ST20xxxx	DB2UTILXCLEAN
ST21xxxx	DB2START
ST22xxxx	DB2STOP
ST23	BCSCLEAN

Offline cloning jobs

This table provides a list of the member names of the offline cloning jobs that are generated and the commands that are run in the jobs.

Table 48. Offline cloning job and member reference

Member name	Command name
ST01xxxx	DB2STOP
ST02	COPY
ST03xxxx	DB2START
ST04	COPYCHECK
ST05	RENAME
ST06xxxx	DB2UPDATE
ST07xxxx	DB2UPDATE (for secondary members of a data sharing group)
ST08xxxx	DB2RBLDBSDS

Table 48. Offline cloning job and member reference (continued)

Member name	Command name
ST09xxxx	DB2LGRNXCLEAN
ST10xxxx	DB2START
ST11xxxx	DB2SQL
ST12xxxx	DB2STOP
ST13xxxx	DB2UTILXCLEAN
ST14xxxx	DB2START
ST15xxxx	DB2STOP
ST23	BCSCLEAN

Jobs for online cloning from a DB2 system level backup

This table provides a list of the member names of the jobs that are generated when online cloning from a system level backup, and the commands that are run in the jobs.

Table 49. Job and member reference for online cloning from a system level backup

Member name	Command name
ST01	DB2GETBACKINFO
ST02	BACKINFO-REFORMAT
ST03	COPY (PGM NONE)
ST04	COPY
ST05	COPYCHECK
ST06	CKZRNTGT
ST07	VOOPTIONS
ST08	RENAME
ST09xxxx	DB2UPDATE
ST10xxxx	DB2UPDATE (for secondary members of a data sharing group)
ST11xxxx	DB2START
ST12xxxx	DB2START (for secondary members of a data sharing group, except for the last secondary member)
ST13xxxx	DB2START (for last secondary member of a data sharing group)
ST14xxxx	DB2STOP (for secondary members of a data sharing group, except for the last secondary member)
ST15xxxx	DB2FIX (DB2)
ST16xxxx	DB2STOP
ST17xxxx	DB2UPDATE
ST18xxxx	DB2START
ST19xxxx	DB2SQL
ST20xxxx	DB2FIX (application)

Table 49. Job and member reference for online cloning from a system level backup (continued)

Member name	Command name
ST21xxxx	DB2STOP
ST22xxxx	DB2RBLDBSDS
ST23xxxx	DB2LGRNXCLEAN
ST24xxxx	DB2XCFCLEAN
ST25xxxx	DB2UTILXCLEAN
ST26xxxx	DB2START
ST27xxxx	DB2STOP
ST28	BCSCLEAN

Jobs for online cloning from a DB2 system level backup using dump tapes as source

This table provides a list of the member names of the jobs that are generated when online cloning from a system level backup using dump tapes as a source, and the commands that are run in the jobs.

Table 50. Job and member reference for online cloning from a system level backup using dump tapes as source

Member name	Command name
ST01	DB2GETBACKINFO
ST02	RESTORE-FROM-DUMPTAPES
ST02RR	RESTORE-FROM-DUMPTAPES(RERUN)
ST03	COPY (PGM NONE)
ST08	RENAME
ST09xxxx	DB2UPDATE
ST10xxxx	DB2UPDATE (for the secondary members of a data sharing group)
ST11xxxx	DB2START
ST12xxxx	DB2START (for secondary members of a data sharing group, except for the last secondary member)
ST13xxxx	DB2START (for last secondary member of a data sharing group)
ST14xxxx	DB2STOP (for secondary members of a data sharing group, except for the last secondary member)
ST15xxxx	DB2FIX (DB2)
ST16xxxx	DB2STOP
ST17xxxx	DB2UPDATE DBD01ONLY
ST18xxxx	DB2START
ST19xxxx	DB2SQL
ST20xxxx	DB2FIX (application)
ST21xxxx	DB2STOP

Table 50. Job and member reference for online cloning from a system level backup using dump tapes as source (continued)

Member name	Command name
ST22xxxx	DB2RBLDBSDS
ST23xxxx	DB2LGRNXCLEAN
ST24xxxx	DB2XCFCLEAN
ST25xxxx	DB2UTILXCLEAN
ST26xxxx	DB2START NORMAL
ST27xxxx	DB2STOP
ST28	BCSCLEAN

Submit the jobs

After the subsystem cloning jobs have been generated, submit the jobs in order.

About this task

Jobs are built with members names of STxx, where xx is an indicator of the order in which the jobs should be submitted. Jobs with the same xx number can be submitted simultaneously. For instance, jobs ST06D9A3 and ST06D9B3 can be run at the same time.

Table space cloning

This section describes the basic steps for table space cloning using the ISPF interface.

Note: Be sure the DDs required for table space cloning have been created before attempting to build a table space cloning profile.

Start the ISPF interface and create a table space cloning profile. After the cloning profile has been created, the Edit DB2 Tablespace Clone Profile menu is displayed, as shown in the following figure:

```

                                Edit DB2 Tablespace Clone Profile
Option ==>

Creator . . . : TWUSR           Name . . . . : TEST PROFILE
Share Option . : UPDATE        Description . .

1 Source and Target DB2 subsystems
2 Source job
3 Target job
4 Report job
5 TCPIP Server job
6 Source TCPIP Server job Required for cross LPAR log apply
  
```

Table space cloning steps summary

The general steps for using the ISPF interface to clone a DB2 table space or index space are described in this topic.

Procedure

1. Ensure the source and target subsystems have been added and properly configured under the Administrative Options option.
2. Create a table space cloning profile.
3. Select or verify the source and target subsystems.
4. Edit and verify the source job settings.
5. Edit and verify the target job settings.
6. Optionally, edit and verify the report job settings.
7. If your system configuration requires the TCP/IP server job and it is not currently active on the target subsystem, edit and verify the TCP/IP server job settings. Refer to “TCP/IP job overview (optional)” on page 170 for detailed information about the TCP/IP server job.
8. If you are using LOG-APPLY processing and your source and target subsystems are on different LPARs, edit and verify the source TCP/IP server job settings. Refer to “Source TCP/IP server job (required when using LOG-APPLY across multiple LPARs)” on page 266 for detailed information about the source TCP/IP server job.
9. Build the table space cloning profile to generate the jobs.
10. Submit the jobs in the correct order.

Select the source and target DB2 subsystems

If source and target subsystems were specified as defaults in the table space cloning default specifications panels under User Settings, those subsystems are used when creating the cloning profile. However, you can verify or change the source and target subsystems using these steps.

About this task

On the Edit DB2 Tablespace Clone Profile menu, select option 1 (Source and Target DB2 subsystems). The Source and Target DB2 subsystems panel is displayed, as shown in the following figure:

```

Source and Target DB2 subsystems
Command ==>

Commands: S - Edit Source DB2 SSID  T - Edit Target DB2 SSID

Creator . . . . : TWUSR           Name . . . . . : TEST PROFILE
Share Option . . : UPDATE         Description . . :

Source SSID . . . . . HAD5         (asterisk to select from list)
IP-VERSION6 . . . . . NO          (Yes/No)
SERVER-IP . . . . . 123.457.78.9
SERVER-PORT . . . . . 5099        (1-65535)
TCPIP-STC-NAME . . . . .

Target SSID . . . . . HAD6         (asterisk to select from list)
IP-VERSION6 . . . . . NO          (Yes/No)
SERVER-IP . . . . . 987.465.123.2
SERVER-PORT . . . . . 5099        (1-65535)
TCPIP-STC-NAME . . . . . TCPIP
LOCATION . . . . .
USERID . . . . .
PASSWORD . . . . .
DEFVCAT . . . . .

```

Procedure

1. Verify that the source SSID is correctly listed in the **Source SSID** field and that the target SSID is correctly listed in the **Target SSID** field.
2. When finished, press PF3 (END) to save and exit. The Edit DB2 Tablespace Clone Profile is displayed.

Edit the source job

If defaults were specified in the table space cloning default specifications panels under User Settings, a table space cloning profile can be created simply by verifying the source and target subsystems and specifying the objects for the LISTDEF command.

About this task

On the Edit DB2 Tablespace Clone Profile menu, select option 2 (Source job). The Setup Source Job menu is displayed, as shown in the following figure:

```

Setup Source Job
Option ==>

Creator . . . . : TWUSR           Name . . . . . : TEST PROFILE
Share Option . . : UPDATE         Description . . :

1 Job card and qualifiers
2 DD Specification
3 SET Command
4 COPY Command
5 HLQDDDF Command
6 XML Object Definition
7 LISTDEF Commands
8 Data Masking Commands

```

Procedure

1. On the Setup Source Job menu, enter 3 (SET Command). The DB2 tablespace clone SET Command panel is displayed.

2. Verify the source SSID is correctly listed in the **LOCAL-SSID** field. Verify or modify other SET command settings as required.
3. Press PF3 (END) to exit.
4. On the Setup Source Job menu, enter 4 (COPY command). The DB2 tablespace clone COPY Command panel is displayed.
5. Verify the target SSID is correctly listed in the **TARGET-DB2 SSID** field. Verify or modify other COPY command settings as required.
6. Press PF3 (END) to exit.
7. On the Setup Source Job menu, enter 7 (LISTDEF Commands). The DB2 tablespace clone LISTDEF Commands panel is displayed.
8. Specify the objects to be copied using this panel.
9. When finished, press PF3 (END) to exit, then press PF3 again. The Edit DB2 Tablespace Clone Profile is displayed.

Verify the target job settings

The job cards and DDs for the target job may have already been specified in the user defaults; verify that the settings are correct.

About this task

On the Edit DB2 Tablespace Clone Profile menu, select option 3 (Target job). The Setup Target Job menu is displayed, as shown in the following figure:

```

Option ==>>                               Setup Target Job

Creator . . . . : TWUSR                     Name . . . . : TEST PROFILE
Share Option . . : UPDATE                   Description . . :

1 Job card
2 DD Specification

```

Select each menu option to verify settings.

Verify the report job settings

The report job is optional; it prints a report based on data in the target job runtime repository. The job cards and DDs for the report job may have already been specified in the user defaults; verify that the settings are correct.

About this task

On the Edit DB2 Tablespace Clone Profile menu, select option 4 (Report job). The Setup Report Job menu is displayed, as shown in the following figure:

```

Option ==>>                               Setup Report Job

Creator . . . . : TWUSR                     Name . . . . : TEST PROFILE
Share Option . . : UPDATE                   Description . . :

1 Job card
2 DD Specification

```

Select each menu option to verify settings.

Verify the TCP/IP server job settings

The TCPIP server job is optional; it facilitates communication between the source job and a target DB2 subsystem on a different z/OS system. The job cards and DDs for the TCPIP job may have already been specified in the user defaults; verify that the settings are correct.

About this task

On the Edit DB2 Tablespace Clone Profile menu, select option 5 (TCPIP job). The Setup TCPIP Server Job menu is displayed, as shown in the following figure:

```

                                Setup TCPIP Server Job
Option ==>

Creator . . . : TWUSR           Name . . . . : TEST PROFILE
Share Option . : UPDATE        Description . :

1 Job card
2 DD Specification
3 SET Command
```

Select each menu option to verify settings.

Verify the source TCP/IP server job settings

The source TCP/IP server job is required only when you are using LOG-APPLY while cloning across multiple LPARs; it facilitates communication between the target job and source DB2 subsystem.

About this task

On the Edit DB2 Tablespace Clone Profile menu, select option 6 (Source TCPIP job). The Setup Source TCPIP Server Job menu is displayed, as shown in the following figure:

```

                                Setup Source TCPIP Server Job
Option ==>

Creator . . . : TWUSR           Name . . . . : TEST PROFILE
Share Option . : UPDATE        Description . :

1 Job card
2 DD Specification
3 SET Command
```

Select each menu option to verify settings.

Build the table space cloning jobs from a profile

Once the profile has been created, build the profile to produce the table space cloning jobs.

Before you begin

Be sure the DDs required for table space cloning have been created before attempting to build a table space cloning profile.

About this task

Begin building the profile on the DB2 Tablespace Clone Profile Display, as shown in the following figure:

```
DB2 Tablespace Clone Profile Display
Command ==>                               Scroll ==> CSR
Commands:      C - Create
Line Commands: B - Build D - Delete E - Edit R - Rename V - View C - Copy

Profile Like . . . *
Creator Like . . . TWUSR*

                                Row 1 of 1
Cmd Name          Creator  Share Option Description
TEST PROFILE     TWUSR    UPDATE
```

Procedure

1. On the DB2 Tablespace Clone Profile Display, enter B next to the profile that you want to build.
2. On the Build DB2 tablespace clone jobs menu, enter 1 to generate source and target jobs and press Enter. The Generate Source and Target Jobs panel is displayed.
3. Specify the data set and member names for the source and target jobs.
4. If desired, select one or more processing options.
5. Press PF3 (END) to continue. If you specified to review the jobs, they are displayed in an ISPF edit session.
6. Press PF3 (END) until you return to the Build DB2 tablespace clone jobs menu.
7. If you want to generate the report job:
 - a. Enter 2 and press Enter. The Generate Report Job panel is displayed.
 - b. Specify the data set and member name for the report job.
 - c. If desired, select one or more processing options.
 - d. Press PF3 (END) to continue. If you specified to review the job, it is displayed in an ISPF edit session.
 - e. Press PF3 (END) until you return to the Build DB2 tablespace clone jobs menu.
8. If you want to generate the TCP/IP server job:
 - a. Enter 3 and press Enter. The Generate TCPIP Server Job panel is displayed.
 - b. Specify the data set and member name for the TCP/IP server job.
 - c. If desired, select one or more processing options.
 - d. Press PF3 (END) to continue. If you specified to review the job, it is displayed in an ISPF edit session.
 - e. Press PF3 (END) until you return to the Build DB2 tablespace clone jobs menu.
9. If you want to generate the source TCP/IP server job (required for LOG-APPLY when the source and target DB2 systems are on different LPARs.):

- a. Enter 4 and press Enter. The Generate Source TCPIP Server Job panel is displayed.
- b. Specify the data set and member name for the source TCP/IP server job.
- c. If desired, select one or more processing options.
- d. Press PF3 (END) to continue. If you specified to review the job, it is displayed in an ISPF edit session.
- e. Press PF3 (END) until you return to the Build DB2 tablespace clone jobs menu.

Results

The jobs have been generated and are in the data sets specified. You can now submit the table space cloning jobs to perform the cloning.

Submit the jobs

After the jobs have been generated, submit the jobs in the correct order.

Procedure

1. Run the optional TCP/IP server job if required.
2. Run the optional source TCP/IP server job, if required.
3. Run the source job.
4. Run the target job.
5. Run the optional report job.

Results

The table space cloning is now complete.

Chapter 22. Cloning table spaces from DB2 Administration Tool

DB2 Cloning Tool integrates with IBM DB2 Administration Tool for z/OS (also referred to as DB2 Admin). Use DB2 Admin to select the table spaces to clone. Then to clone the spaces, invoke DB2 Cloning Tool directly from the DB2 Admin interface.

Before you begin

Ensure that the DB2 Cloning Tool customization tasks that are described in Chapter 4, “Customizing DB2 Cloning Tool,” on page 51 and Chapter 5, “After customizing DB2 Cloning Tool using Tools Customizer,” on page 71 have been completed.

Procedure

1. Run the DB2 Admin CLIST.
2. Select table spaces to clone using the Databases panel, the Table Spaces panel, or the Tables, Views, and Aliases panel.
3. Type the **CT** command on the command line and press Enter. All objects listed on the displayed panel are selected for cloning.
4. Optional: On the Clone Tables panel, refine the list of spaces to be cloned by using the available DB2 Admin commands.
5. On the Clone Tables panel, type **CONTINUE** on the command line and press Enter.
6. Create a new cloning profile or select an existing DB2 Cloning Tool cloning profile.
 - To create a new profile, complete the following steps:
 - a. On the Specify clone profile window, type **1** in the **Select an action** field, and press Enter.
 - b. On the Enter New DB2 Tablespace Clone Profile Options window, enter a new profile name and share option. Press Enter.
 - To use an existing profile, complete the following steps:
 - a. On the Specify clone profile window, type **2** in the **Select an action** field, and press Enter.
 - b. On the DB2 Tablespace Clone Profile Selection panel, select a profile, and press Enter.
 - c. You are prompted to add the table spaces that you selected using DB2 Admin to the existing profile, or replace the existing table spaces in the profile with those you selected using DB2 Admin. Press Enter.
7. On the Edit DB2 Tablespace Clone Profile panel, type **1** to edit the source job, and press Enter.
8. On the Setup Source Job panel, type **4** to edit the COPY command information, and press Enter.
9. On the DB2 tablespace clone COPY Command panel, enter the target subsystem in the **TARGET-DB2 SSID** field, or if using an existing profile, verify that the correct target subsystem has been specified. If the target has

not been specified, type the target subsystem name in the field. To select the subsystem from a list, type * in the field, and press Enter. If the subsystem does not exist:

- a. Type the target subsystem name in the **TARGET-DB2 SSID** field.
 - b. Type **I** in the command line and press Enter.
 - c. Specify the required information for the target subsystem that is described in the topic “Configuring DB2 subsystems” on page 276.
10. On the Setup Source Job panel, type **7** to review LISTDEF command information, and press Enter.
 11. On the DB2 Tablespace Clone LISTDEF Commands panel, review the list of table spaces that were selected using DB2 Admin and modify it if required.
 12. Press PF3 until the Edit DB2 Tablespace Clone Profile panel is displayed.
 13. Continue with table space cloning process, following the steps starting with the topic “Edit the source job” on page 329.

Chapter 23. DB2 Cloning Tool Subsystem Cloning commands

All DB2 Cloning Tool Subsystem Cloning commands are invoked by running the main program CKZ00010. Functionality is selected by specifying the appropriate DB2 Cloning Tool command and parameters.

Required JCL varies with each command. The major factor in choosing whether COPY and RENAME will be in the CKZ00010 execution in separate steps, or in separate jobs, is whether resumption of source volume access will begin after the COPY step, or after the RENAME step.

The format of the DB2 Cloning Tool journal records can change between releases, so a journal data set created by one release of DB2 Cloning Tool should not be used by a different release of DB2 Cloning Tool. If you need to use an older release of the journal, use the JRNLUPGRADE command.

The DB2 Cloning Tool commands are listed in the following table and also explained in detail in separate topics.

Table 51. DB2 Cloning Tool commands overview

DB2 Cloning Tool Subsystem Cloning Command	Description
"BACKINFO-REFORMAT" on page 338	Optional command that takes a backinfo data set created by the DB2GETBACKINFO command and reformats it for use by subsequent COPY commands and optionally a VOOPTIONS command.
"BCSCLEAN" on page 345	Optional command to delete target catalog entries from previous execution.
"COPY" on page 350	Initiates volume copies, and in parallel, backs up the source ICF catalogs that point to data sets on the source volumes being cloned.
"COPYCHECK" on page 371	Optional command to wait for completion of COPY events, or withdraw from them.
"DB2ALTERBSDS" on page 374	Optional command to alter the contents of a target BSDS beyond what DB2UPDATE does.
"DB2FIX" on page 379	Optional command to correct DB2 page spaces that are in restricted status.
"DB2GETBACKINFO" on page 385	Optional command to issue HSM LIST COPYPOOL commands to determine the available backups from DB2 BACKUP SYSTEM and read the results to determine the source and backup volume pairs to be used.
"DB2LGRNXCLEAN" on page 390	Optional command to clean out DB2 SYSLGRNX as part of removing other members.
"DB2RBLDBSDS" on page 393	Optional command to rebuild a DB2 BSDS to remove other members or to make it non-data sharing.
"DB2SETLOG" on page 396	Optional command to suspend or resume a DB2 subsystem.
"DB2SQL" on page 399	Optional command to update the DB2 catalog.
"DB2START" on page 404	Optional command to start a DB2 subsystem.

Table 51. DB2 Cloning Tool commands overview (continued)

DB2 Cloning Tool Subsystem Cloning Command	Description
"DB2STOP" on page 409	Optional command to stop a DB2 subsystem.
"DB2UPDATE" on page 411	Optional command to update DB2 to reflect renamed data sets.
"DB2UTILXCLEAN" on page 420	Optional command to remove utility information from the target DB2 subsystem.
"DB2XCFCLEAN" on page 424	Optional command to clean out DB2 XCF structures and group members.
"FINDUCATS" on page 426	Optional command to locate catalogs involved with source volume data sets.
"JRNLUPLGRADE" on page 428	Optional command to upgrade a journal created by a prior release of DB2 Cloning Tool.
"ONLINECLIP" on page 431	Optional command to re-label the target volume(s) when the source volume label was copied but the UCB field still points to the target volume label. This can occur when TSO FCESTABL was used.
"RENAME" on page 433	Renames and catalogs data sets on target volumes.
"RESTORE-FROM-DUMPTAPES" on page 450	Optional command that allows DB2 Cloning Tool to use the backups on tape from a DB2 BACKUP SYSTEM DUMP utility as the source for cloning.
"UCATOPTIONS" on page 456	Optional command that will either list the user catalog pairs from the DB2 Cloning Tool journal or allow the target user catalog name(s) to be changed.
"VARYOFF" on page 459	Optional command to vary target or source volumes offline.
"VARYON" on page 463	Optional command to vary target or source volumes online.
"VOLOPTIONS" on page 468	Optional command to use when the COPY command is run at one site and the RENAME command is run at another.

BACKINFO-REFORMAT

This command is not required. BACKINFO-REFORMAT will take a backinfo data set created by the DB2GETBACKINFO command and reformat it for use by subsequent COPY commands and optionally a VOLOPTIONS command.

This command is used as part of the process to clone from a DB2 BACKUP SYSTEM backup. A detailed description of this cloning process can be found in "Cloning scenarios" on page 535.

The input backinfo data set consists of two types of records. The VOLMAP records identify the source volumes and their corresponding backup volumes. The UCAT records identify the source ICF catalogs and the source volumes they reside on.

The output FROM-VOLSER-DDN data set consists of a list of the backup volumes. This data set will be used in a subsequent COPY to a FROM-VOLSER-DDN keyword.

The output VOLPAIRS-DDN data set consists of source and backup volume pairs. This data set will be used in a subsequent COPY to a VOLPAIRS-DDN keyword.

The output USERCATALOGS-DDN data set consists of source ICF catalog with volser and target ICF catalog pairs. This data set will be used in a subsequent COPY to a USERCATALOGS-DDN keyword. The USERCATALOGS keyword is used to specify source and target pairs of ICF catalogs. The source ICF catalog volser will be filled in based on the UCAT records in the backinfo data set.

The output USERSGDEFS-DDN data set consists of a list of the backup volumes and the storage group names of the corresponding source volumes. This data set will be used in a subsequent COPY to a USERSGDEFS-DDN keyword. To use this data set in COPY, the COPY command must include a USERSGDEFS-OFFSETS(VOLSER(1) SGNAME(8) INCLEXCL(18)) keyword. This allows the FROM-USER-STORAGEGROUP keyword to be used in the COPY command, so the backup volumes can be paired to the target volumes by using the storage group names of the source and target volumes.

The output VOLOPTIONS-CMD-DDN data set consists of a generated VOLOPTIONS command. This data set will be used in a subsequent VOLOPTIONS command. The generated command will be:

```
VOLOPTIONS xxxx -  
    JOURNAL-DDN(JOURNAL)
```

Where *xxxx* will be LIST or UNCLIP. UNCLIP will be inserted if CLIP-IF-OFFLINE(Y) is specified and any backup volumes were clipped in this run. Otherwise, LIST will be inserted. The VOLOPTIONS-CMD-DDN keyword is intended to be used when the backup volumes are offline and have an internal volser of their corresponding source volume. BACKINFO-REFORMAT will clip the backup volumes and vary them online. After the backup volumes have been copied to the target volumes the VOLOPTIONS UNCLIP command can be used to return the backup volumes to their original state (offline with internal volser of source volume).

When CLIP-IF-OFFLINE(Y) and the backinfo VOLMAP record does not give a unique volser for a backup volume, BACKINFO-REFORMAT will generate a volser to use for the backup volume based on the masks in the VOLSER-RENAME-MASKS keyword.

BACKINFO-REFORMAT command syntax

BACKINFO-REFORMAT

Required Keywords:

```
BACKINFO-DDN( ddname )  
{ FROM-VOLSER-DDN( ddname ) | USERSGDEFS-DDN( ddname ) }  
VOLPAIRS-DDN( ddname )
```

Optional Keywords:

```
CLIP-IF-OFFLINE( N | Y )  
RESUME  
SIMULATE  
USERCATALOGS( sourcecat1 targetcat1 ...[ , sourcecatn targetcatn ] )  
USERCATALOGS-DDN( ddname )  
VOLOPTIONS-CMD-DDN( ddname )
```

Required only if CLIP-IF-OFFLINE is specified:

VOLSER-RENAME-MASKS(*sourcemask1 backupmask1 ...[, sourcemaskn backupmaskn]*)

VOLSER-RENAME-MASKS Considerations

Oldvalue Syntax

The old value filter mask is used to select the source volume volser values against which to apply the new value mask. The allowable filter characters are shown in the following table:

Table 52. Filter characters allowed for oldvalue filter masks:

Character	Description
*	A single asterisk represents 0 to nn characters of any value.
%	A percent sign represents one non-blank character.
!	An exclamation point represents one national character. @ # \$
<	A lesser-than sign represents one non-numeric character, national symbols included.
>	A greater-than sign represents one numeric character.

For example, Old value filter mask = SRC* would match source volume volser = SRC001.

For information about filters and ACS masks, refer to the topic “Filtering pattern masks” on page 23.

Newvalue Syntax

The new value filter mask is used to rename the source volume volser value selected by the old value filter mask. The allowable filter characters are shown in the following table:

Table 53. Filter characters allowed for newvalue filter masks

Character	Description
*	A single asterisk represents 0 to nn characters of any value. The single asterisk may only be used as the last character of the mask.
%	A percent sign represents one non-blank character.

For example: Source volume volser = SRC001, New value filter mask = BKP*, New backup volume volser = BKP001.

For information about filters and ACS masks, refer to the topic “Filtering pattern masks” on page 23.

BACKINFO-REFORMAT command and keyword definitions

Required keywords are described first, followed by optional keywords.

Required BACKINFO-REFORMAT keywords

BACKINFO-REFORMAT

Optional command to take a backinfo data set created by the DB2GETBACKINFO command and reformat it for use by subsequent COPY commands and optionally a VOLOPTIONS command.

- Required: No

- Restrictions: None

BACKINFO-DDN(*ddname*)

This parameter specifies the DD name that points to a file containing the backinfo data. BACKINFO-DDN must have an LRECL of 80.

- Default: None
- Required: Yes
- Restrictions: None

FROM-VOLSER-DDN(*ddname*)

This parameter specifies the DD name that points to a file where the from volser information will be written.

- Default: None
- Required: Yes
- Restrictions: Mutually exclusive with USERSGDEFS-DDN.

USERSGDEFS-DDN(*ddname*)

This parameter specifies the DD name that points to a file where the backup volume and source storage group information will be written.

- Default: None
- Required: Yes
- Restrictions: Mutually exclusive with FROM-VOLSER-DDN.

VOLPAIRS-DDN(*ddname*)

This parameter specifies the DD name that points to a file where the volpairs information will be written.

- Default: None
- Required: Yes
- Restrictions: None

Optional BACKINFO-REFORMAT keywords

CLIP-IF-OFFLINE(N | Y)

This parameter specifies that DB2 Cloning Tool may clip offline backup volumes. If CLIP-IF-OFFLINE is not specified, or specified with N and a backup volume is offline, the process fails.

If CLIP-IF-OFFLINE(Y) is specified, offline backup volumes will be clipped and varied online. The offline backup volumes are expected to have an internal volser that matches their corresponding source volume volser. The clipped volser of a backup volume is determined by the value given in the backinfo data set if unique or the masks specified in the VOLSER-RENAME-MASKS keyword.

- Default: N
- Required: No
- Restrictions: None

RESUME

RESUME specifies that processing should resume for any volumes that failed to be completely processed by the previous run.

Prior to running with RESUME, the problem that caused a volume to not be completely processed should be resolved

- Default: None
- Required: No

- Restrictions: Only applies when CLIP-IF-OFFLINE(Y) is specified.

SIMULATE

SIMULATE will verify the syntax, determine the volumes to be processed, and display what action would have been taken, but will not change any volume serials with ICKDSF or vary any volumes online.

- Default: None
- Required: No
- Restrictions: Only applies if CLIP-IF-OFFLINE(Y) is specified

USERCATALOGS (*sourcecat1 targetcat1, ..., sourcecatn targetcatn*)

This parameter specifies source catalogs that data sets from source volumes are cataloged in, and the corresponding target catalog that renamed volume data sets are to be cataloged in. The specified source catalogs must also be in the backinfo data set in a UCAT record.

- Default: None
- Required: No
- Restrictions: Can only be specified if the backinfo data set has UCAT records.

USERCATALOGS-DDN (*ddname*)

This parameter specifies the DD name that points to a file where the usercatalogs information will be written.

- Default: None
- Required: Required if USERCATALOGS specified.
- Restrictions: Required if USERCATALOGS specified.

VOLOPTIONS-CMD-DDN (*ddname*)

This parameter specifies the DD name that points to a file where the VOLOPTIONS command will be written.

The VOLOPTIONS command will have the LIST option if no volumes are clipped by BACKINFO-REFORMAT and will have the UNCLIP option if volumes are clipped by BACKINFO-REFORMAT.

This parameter is intended to be used when the backup volumes are offline and are being clipped and varied online for a COPY command, after which the backup volumes will be varied offline and clipped back to their corresponding source volume volser.

- Default: None
- Required: No
- Restrictions: None

VOLSER-RENAME-MASKS (*sourcemark1 backupmask1, ... sourcemarkn backupmaskn*)

VOLSER-RENAME-MASKS are specified in 'oldvolser' 'newvolser' pairs. VOLSER-RENAME-MASKS are processed in order. The first hit of the source volser is the one that is used for the backup volser.

- Default: None
- Required: Required if CLIP-IF-OFFLINE(Y) is specified.
- Restrictions: Only applies when CLIP-IF-OFFLINE(Y) is specified.

BACKINFO-REFORMAT step JCL examples

| Two BACKINFO-REFORMAT step JCL examples are included: an example that
| uses FROM-VOLSER-DDN to put out the backup volume information, and an

example that uses USERSGDEFS-DDN to put out the backup volumes and the corresponding source storage group name information. Sample JCL can be found in the installation library SCKZJCL in member CKZBKIRF.

BACKINFO-REFORMAT Step JCL – example that uses FROM-VOLSER-DDN to put out backup volume information

For completeness and to illustrate where JCL parameters and DB2 Cloning Tool control statements must match, the following JCL includes sample DB2 Cloning Tool control statements. Complete BACKINFO-REFORMAT command control statement syntax is documented in the topic “BACKINFO-REFORMAT command syntax” on page 339.

The BACKINFO-REFORMAT step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

1 //??????? JOB , 'CKZ BACKINFO-REF', CLASS=A, MSGCLASS=X
2 //S0      EXEC PGM=IDCAMS
3 //SYSPRINT DD  SYSOUT=*
4 //SYSIN   DD  *
5 1      DEL CKZ.WRK.VOLPAIRS
6 2      DEL CKZ.WRK.FRVOLSER
7 3      DEL CKZ.WRK.UCATS
8      SET MAXCC=0
9 4 //S1      EXEC PGM=CKZ00010, REGION=8M
10 5 //STEPLIB DD  DSN=HLQ?.SCKZLOAD, DISP=SHR
11 6 //CKZINI  DD  DSN=HLQ?.SCKZPARM(CKZINI), DISP=SHR
12 7 //CKZPRINT DD  SYSOUT=*
13 //SYSUDUMP DD  SYSOUT=*
14 8 //BACKINFO DD  DISP=SHR, DSN=CKZ.WRK.BACKINFO
15 9 //VOLPAIRS DD  DSN=CKZ.WRK.VOLPAIRS,
16 //          DISP=(,CATLG), UNIT=SYSALLDA,
17 //          SPACE=(CYL, (1,1))
18 10 //FRVOLSER DD  DSN=CKZ.WRK.FRVOLSER,
19 //          DISP=(,CATLG), UNIT=SYSALLDA,
20 //          SPACE=(CYL, (1,1))
21 11 //UCATS   DD  DSN=CKZ.WRK.UCATS,
22 //          DISP=(,CATLG), UNIT=SYSALLDA,
23 //          SPACE=(CYL, (1,1))
24 //CKZIN    DD  *
25      BACKINFO-REFORMAT          -
26 8      BACKINFO-DDN(BACKINFO)   -
27 9      VOLPAIRS-DDN(VOLPAIRS)   -
28 10     FROM-VOLSER-DDN(FRVOLSER) -
29 11     USERCATALOGS-DDN(UCATS)  -
30 12     USERCATALOGS(           -
31         USERCAT.SRC01 USERCAT.TGT01 -
32         USERCAT.SRC02 USERCAT.TGT02 -
33     )

```

1. Deletion of volpairs data set in anticipation of allocating new for each execution.
2. Deletion of from volser data set in anticipation of allocating new for each execution.
3. Deletion of usercatalogs data set in anticipation of allocating new for each execution.
4. Execution of DB2 Cloning Tool main program.
5. DB2 Cloning Tool SCKZLOAD library (must be authorized).
6. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the DB2 Cloning Tool programs.

7. DD for CKZPRINT output.
8. The backinfo data set. This data set is used to pass information to the BACKINFO-REFORMAT command from the DB2GETBACKINFO command. In the sample JCL, the control statement BACKINFO-DDN(BACKINFO) specifies that a DD statement with the name BACKINFO is used.
9. The volpairs data set that is created by BACKINFO-REFORMAT and will be used by a subsequent DB2 Cloning Tool COPY command. In the sample JCL, the control statement VOLPAIRS-DDN(VOLPAIRS) specifies that a DD statement with the name VOLPAIRS is used.
10. The from volser data set that is created by BACKINFO-REFORMAT and will be used by a subsequent DB2 Cloning Tool COPY command. In the sample JCL, the control statement FROM-VOLSER-DDN(FRVOLSER) specifies that a DD statement with the name FRVOLSER is used.
11. The usercatalogs data set that is created by BACKINFO-REFORMAT and will be used by a subsequent DB2 Cloning Tool COPY command. In the sample JCL, the control statement USERCATALOGS-DDN(UCATS) specifies that a DD statement with the name UCATS is used.
12. The ICF user catalog pairs that will be validated and written to the USERCATALOGS-DDN. In the sample JCL, the control statement USERCATALOGS-DDN(UCATS) specifies that a DD statement with the name UCATS is used.

BACKINFO-REFORMAT Step JCL – example that uses USERSGDEFS-DDN to put out backup volume and corresponding source storage group name information

For completeness and to illustrate where JCL parameters and DB2 Cloning Tool control statements must match, the following JCL includes sample DB2 Cloning Tool control statements. Complete BACKINFO-REFORMAT command control statement syntax is documented in the topic “BACKINFO-REFORMAT command syntax” on page 339.

The BACKINFO-REFORMAT step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

1      //??????? JOB , 'CKZ BACKINFO-REF' ,CLASS=A,MSGCLASS=X
2      //S0      EXEC PGM=IDCAMS
3      //SYSPRINT DD SYSOUT=*
4      //SYSIN   DD *
5      1      DEL CKZ.WRK.VOLPAIRS
6      2      DEL CKZ.WRK.USRSGDEF
7      3      DEL CKZ.WRK.UCATS
8          SET MAXCC=0
9      4 //S1      EXEC PGM=CKZ00010,REGION=8M
10     5 //STEPLIB DD DSN=HLQ?.SCKZLOAD,DISP=SHR
11     6 //CKZINI  DD DSN=HLQ?.SCKZPARM(CKZINI),DISP=SHR
12     7 //CKZPRINT DD SYSOUT=*
13     //SYSUDUMP DD SYSOUT=*
14     8 //BACKINFO DD DISP=SHR,DSN=CKZ.WRK.BACKINFO
15     9 //VOLPAIRS DD DSN=CKZ.WRK.VOLPAIRS,
16         //      DISP=(,CATLG),UNIT=SYSALLDA,
17         //      SPACE=(CYL,(1,1))
18     10 //USRSGDEF DD DSN=CKZ.WRK.USRSGDEF,
19         //      DISP=(,CATLG),UNIT=SYSALLDA,
20         //      SPACE=(CYL,(1,1))
21     11 //UCATS   DD DSN=CKZ.WRK.UCATS,
22         //      DISP=(,CATLG),UNIT=SYSALLDA,
23         //      SPACE=(CYL,(1,1))

```

```

|          //CKZIN  DD  *
|          BACKINFO-REFORMAT          -
| 8      BACKINFO-DDN(BACKINFO)      -
| 9      VOLPAIRS-DDN(VOLPAIRS)      -
|10     USERSGDEFS-DDN(USRSGDEF)     -
|11     USERCATALOGS-DDN(UCATS)      -
|12     USERCATALOGS(                -
|          USERCAT.SRC01 USERCAT.TGT01 -
|          USERCAT.SRC02 USERCAT.TGT02 -
|          )

```

1. Deletion of volpairs data set in anticipation of allocating new for each execution.
2. Deletion of USERSGDEFS data set in anticipation of allocating new for each execution.
3. Deletion of usercatalogs data set in anticipation of allocating new for each execution.
4. Execution of DB2 Cloning Tool main program.
5. DB2 Cloning Tool SCKZLOAD library (must be authorized).
6. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the DB2 Cloning Tool programs.
7. DD for CKZPRINT output.
8. The backinfo data set. This data set is used to pass information to the BACKINFO-REFORMAT command from the DB2GETBACKINFO command. In the sample JCL, the control statement BACKINFO-DDN(BACKINFO) specifies that a DD statement with the name BACKINFO is used.
9. The volpairs data set that is created by BACKINFO-REFORMAT and will be used by a subsequent DB2 Cloning Tool COPY command. In the sample JCL, the control statement VOLPAIRS-DDN(VOLPAIRS) specifies that a DD statement with the name VOLPAIRS is used.
10. The backup volser and source storage group data set that is created by BACKINFO-REFORMAT and will be used by a subsequent DB2 Cloning Tool COPY command. In the sample JCL, the control statement USERSGDEFS-DDN(USRSGDEF) specifies that a DD statement with the name USRSGDEF is used.
11. The usercatalogs data set that is created by BACKINFO-REFORMAT and will be used by a subsequent DB2 Cloning Tool COPY command. In the sample JCL, the control statement USERCATALOGS-DDN(UCATS) specifies that a DD statement with the name UCATS is used.
12. The ICF user catalog pairs that will be validated and written to the USERCATALOGS-DDN. In the sample JCL, the control statement USERCATALOGS-DDN(UCATS) specifies that a DD statement with the name UCATS is used.

BCSCLEAN

This command is not required. BCSCLEAN is intended for situations where the target catalog may be used for data sets other than those involved with a DB2 Cloning Tool process.

BCSCLEAN and its keywords support three distinct types of catalog cleanup:

- Uncatalog files in a target catalog that were renamed via the RENAME command.

- Uncatalog other files in a target catalog and on target volumes via the CLEANUP-CATALOG-ORPHANS keyword.
- Delete other files in a target catalog by data set name mask via the CLEANUP-CATALOG-DSNMASKS keyword.

For more information about CLEANUP-CATALOG-ORPHANS and CLEANUP-CATALOG-DSNMASKS, refer to the keyword descriptions and the step JCL example in the topic “BCSCLEAN Step JCL - example to clean up target user catalog entries” on page 349.

If an empty target catalog is possible, rather than use BCSCLEAN, the target catalog can simply be deleted and re-defined prior to the DB2 Cloning Tool COPY step. If the redefined catalog is not on the same volume it was on prior to the delete, special care must be taken to inform all the catalog address spaces of its new location. IBM informational APAR II13354 details the steps necessary to ensure all sharing systems can access the catalog.

BCSCLEAN deletes (with no scratch) all catalog entries created in a target catalog by a previous RENAME step. BCSCLEAN is intended to delete target catalog entries created from a previous run of the DB2 Cloning Tool process that may be orphaned as a result of target volume contents being replaced.

Specifically, BCSCLEAN addresses situations where a data set used in a previous application cycle no longer exists in the current application cycle.

Additionally, for persistent data sets (application data sets that exist in every cycle), by emptying the target catalog (delete all entries from a previous run of the DB2 Cloning Tool process), the RECATALOG option of the RENAME command can be omitted, and any existing catalog entry can be treated as a true error.

BCSCLEAN command syntax

BCSCLEAN commands syntax.

BCSCLEAN

Required keywords:

{ JOURNAL-DSN(*data set name*) | JOURNAL-DDN(*ddname*) }

Optional keywords:

CLEANUP-CATALOG-DSNMASKS(*dsnmask1* , ... *dsnmaskn*)
 CLEANUP-CATALOG-ORPHANS
 SIMULATE

BCSCLEAN command and keyword definitions

Required keywords are described first, followed by optional keywords.

BCSCLEAN

Optional command to delete target catalog entries from previous execution.

- Required: No
- Restrictions: None

JOURNAL-DSN (*data set name*)

or JOURNAL-DDN (*ddname*)

This parameter supplies either the data set name of the DB2 Cloning Tool journal file, or a DD name assumed via the JCL to point at a journal data set.

If multiple DB2 Cloning Tool setups are used for different volume groups, DO NOT use the same journal data set. Each DB2 Cloning Tool 'application' needs a different journal data set.

The journal is used to pass information between DB2 Cloning Tool steps (e.g., from the COPY step to the RENAME step). Therefore, as noted in the JCL comments, it must be cataloged in the COPY step when first created, and referenced as OLD in subsequent steps.

Do not delete the data set in the last step, in case restarts or reruns need journal information. It is preferable to delete and replace the data set only at the fresh start of a COPY.

- Default: None
- Required: Yes
- Restrictions: None

CLEANUP-CATALOG-DSNMASKS(*dsnmask1* , ... *dsnmaskn*)

Use this parameter to delete and uncatalog data sets that are not on a target volume, but are in a target user catalog, and as a result of the clone are no longer required. For example, there may be DB2 archive logs that were not part of the cloning process and are not needed on the target. Specify one or more data set name masks from most likely to least likely to match. The masks are checked against the list of target volumes in the journal. The first match ends the comparison. The following types of data sets are uncataloged and deleted; any other data set types are ignored:

- VSAM clusters: If the VSAM cluster name matches any individual data set name mask, the cluster is deleted and scratched. Note: Only the VSAM cluster name is used in the comparison; the VSAM data and index names are not compared.
- Non-VSAM data sets are deleted and scratched if the data set name matches any individual mask.
- GDGs: Only GDS files within a GDG are considered for deletion. Each GDS is a unique non-VSAM file and is individually considered for deletion (as above). A GDS is deleted and scratched if its data set name matches any individual mask. The GDG base file itself is not uncataloged or deleted; the user must manually delete the GDG base.

In some cases, deleting migrated files may cause HSM (or equivalent software) to attempt to recall the file, before delete processing can proceed.

An optional audit log file can be specified to capture a list of files deleted during CLEANUP-CATALOG-DSNMASKS processing. To specify the log file, include a DD named LOGFILE, defined as SYSOUT=*, or as a data set with DSORG=PS and LRECL=80, or as a PDS member with the same DCB characteristics). All files that are processed for deletion are logged, including failures for objects not found or other errors.

- Default: None
- Required: No
- Restrictions: None

CLEANUP-CATALOG-ORPHANS

Use this parameter to uncatalog data sets that are no longer on the target volume after the cloning process, but are still in the target user catalog. For example, there may be non-VSAM data sets or VSAM clusters that existed on the target volume before the clone, but were not part of the cloning process. The data sets are checked against the list of target volumes in the journal to determine whether the data set should be uncataloged. If the

data set is on multiple volumes, all must be target volumes as defined in the journal in order to be deleted. However, candidate volumes are ignored. The following types of data sets are uncataloged; any other data set types are ignored:

- VSAM clusters: All volumes for the VSAM data and index components must be target volumes as defined in the journal file. Otherwise, the file is bypassed and logged with first failing volume and associated data or index component.
- Non-VSAM files: All volumes must be target volumes as defined in the journal file. Otherwise, the file is bypassed and logged with first failing volume.
- GDGs: GDS files within a GDG are considered for deletion. Each GDS is a unique non-VSAM file and is individually considered for deletion (as above). The GDG base file itself is not uncataloged or deleted; the user must manually delete the GDG base.

Note: For each data set, CLEANUP-CATALOG-ORPHANS processing occurs before CLEANUP-CATALOG-DSNMASKS processing. This impacts data sets that could potentially match both keywords (those that are on a target volume and also match a supplied *dsnmask* value, as follows:

- Because a match on any target volume means the data set was deleted by the cloning process, a data set that could match both keywords is processed only by CLEANUP-CATALOG-ORPHANS, and only uncataloged (no delete from the volume is attempted).
- No CLEANUP-CATALOG-DSNMASKS processing is required for such data sets. This can increase efficiency, especially when many data set name masks are supplied.

An optional audit log file can be specified to capture a list of files deleted during CLEANUP-CATALOG-ORPHANS processing. To specify the log file, include a DD named LOGFILE, defined as SYSOUT=*, or as a data set with DSORG=PS and LRECL=80, or as a PDS member with the same DCB characteristics. All files that are processed for deletion are logged, including failures for objects not found or other errors.

- Default: None
- Required: No
- Restrictions: None

SIMULATE

Include this parameter to generate a list of data sets and information about whether they will be uncataloged or deleted during a subsequent non-SIMULATE run. No updates of any kind are performed. Output is generated in CKZPRINT with other BCSCLEAN output.

It is possible that the number of files identified in a SIMULATE run as candidates for deletion differs from the number actually removed by CLEANUP-CATALOG-ORPHANS or CLEANUP-CATALOG-DSNMASKS during the normal (non-SIMULATE) run. In a normal run, standard BCSCLEAN processing occurs first, which might remove some catalog entries that CLEANUP-CATALOG-ORPHANS and CLEANUP-CATALOG-DSNMASKS processing then will not encounter. In a SIMULATE run, the two separate BCSCLEAN processes each review the exact same user catalog records, because no deletes or updates are performed.

- Default: None
- Required: No

- Restrictions: None

BCSCLEAN step JCL examples

This topic contains two examples of BCSCLEAN step JCL.

BCSCLEAN Step JCL example

For completeness and to illustrate where JCL parameters and DB2 Cloning Tool control statements must match, the following JCL includes sample DB2 Cloning Tool control statements. Sample JCL can be found in the installation library SCKZJCL in member CKZBCLN.

Note that the 'BCSRECS' data set specified in the JCL contains the list of catalog entries that were put there by the prior RENAME and are to be deleted.

The BCSCLEAN step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

      //?????? JOB , 'CKZ BCSCLEAN', CLASS=A, MSGCLASS=X
1 //S1      EXEC PGM=CKZ00010, REGION=8M
2 //STEPLIB DD DSN=HLQ?.SCKZLOAD, DISP=SHR
3 //CKZINI  DD DSN=HLQ?.SCKZPARM(CKZINI), DISP=SHR
4 //BCSRECS DD DSN=CKZ.WRK.BCSRECS, DISP=SHR
5 //CKZPRINT DD SYSOUT=*
  //SYSUDUMP DD SYSOUT=*
6 //JOURNAL DD DSN=CKZ.JRNL, DISP=OLD
  //CKZIN   DD *
      BCSCLEAN
6      JOURNAL-DDN(JOURNAL)
  //*
```

1. Execution of DB2 Cloning Tool main program.
2. DB2 Cloning Tool SCKZLOAD library must be authorized.
3. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the DB2 Cloning Tool programs.
4. DD for BCSRECS. The BCSRECS data set contains the names of data sets cataloged in a previous RENAME step. The names contained in this data set determine the catalog entries to be deleted by BCSCLEAN.
5. DD for CKZPRINT output.
6. Journal data set used by DB2 Cloning Tool commands. This VSAM data set is used to pass information between DB2 Cloning Tool steps, and to log information derived from the step executions. The journal file is allocated and cataloged in the COPY step and used as input and output by the RENAME step. The specified data set name must match the data set allocated in the COPY step. If multiple DB2 Cloning Tool setups are used for different groups of volumes, take care in specifying the journal data sets for each setup. In the sample JCL, the control statement JOURNAL-DDN(JOURNAL) specifies that a DD with the name JOURNAL is being used rather than a data set name.

BCSCLEAN Step JCL - example to clean up target user catalog entries

For completeness and to illustrate where JCL parameters and DB2 Cloning Tool control statements must match, the following JCL includes sample DB2 Cloning Tool control statements.

Note that the 'BCSRECS' data set specified in the JCL contains the list of catalog entries that were put there by the prior RENAME and are to be deleted.

The BCSCLEAN step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

      //??????? JOB , 'CKZ BCSCLEAN', CLASS=A, MSGCLASS=X
1 //S1      EXEC PGM=CKZ00010, REGION=8M
2 //STEPLIB DD DSN=HLQ?.SCKZLOAD, DISP=SHR
3 //CKZINI  DD DSN=HLQ?.SCKZPARAM(CKZINI), DISP=SHR
4 //BCSRECS DD DSN=CKZ.WRK.BCSRECS, DISP=SHR
5 //CKZPRINT DD SYSOUT=*
  //SYSUDUMP DD SYSOUT=*
6 //JOURNAL DD DSN=CKZ.JRNL, DISP=OLD
  //CKZIN   DD *
      BCSCLEAN
7      CLEANUP-CATALOG-ORPHANS
8      CLEANUP-CATALOG-DSNMASKS(DB1T.ARCHLOG*.***)
6      JOURNAL-DDN(JOURNAL)
  //*
```

1. Execution of DB2 Cloning Tool main program.
2. DB2 Cloning Tool SCKZLOAD library must be authorized.
3. DD for CKZINI, SCKZPARAM member. The CKZINI member of the HLQ?.SCKZPARAM library provides variables to the DB2 Cloning Tool programs.
4. DD for BCSRECS. The BCSRECS data set contains the names of data sets cataloged in a previous RENAME step. The names contained in this data set determine the catalog entries to be deleted by BCSCLEAN.
5. DD for CKZPRINT output.
6. Journal data set used by DB2 Cloning Tool commands. This VSAM data set is used to pass information between DB2 Cloning Tool steps, and to log information derived from the step executions. The journal file is allocated and cataloged in the COPY step and used as input and output by the RENAME step. The specified data set name must match the data set allocated in the COPY step. If multiple DB2 Cloning Tool setups are used for different groups of volumes, take care in specifying the journal data sets for each setup. In the sample JCL, the control statement JOURNAL-DDN(JOURNAL) specifies that a DD with the name JOURNAL is being used rather than a data set name.
7. CLEANUP-CATALOG-ORPHANS parameter that specifies to uncatalog data sets that would no longer be on the target volume after the cloning process, but would still be in the target user catalog.
8. CLEANUP-CATALOG-DSNMASKS parameter that specifies to delete and uncatalog data sets that are not on a target volume, but are in a target user catalog. In this case, each DB1T.ARCHLOG*.** data set will be checked against the list of target volumes in the journal to determine whether the data set is to be deleted and uncataloged.

COPY

This command is required. COPY invokes volume copies via FlashCopy or SnapShot if the DATA-MOVER(PGM(ADRDSSU)) is specified, or invokes volume copies via TimeFinder/Clone Mainframe Snap Facility's volume level support if the DATA-MOVER(PGM(EMCSNAP)) is specified, or assumes copies have been created by the user if DATA-MOVER(PGM(NONE)) is specified. In all cases, COPY captures catalog data pertaining to source volume data sets.

Important: The ICF catalog backup can be postponed until after COPY by using the USERCATALOGS-NOBACKUP keyword. However, if you choose to postpone the backup, the source ICF catalog information being backed up needs to be in synchronization with the contents of the copied volumes. If USERCATALOGS-NOBACKUP is used (so the source ICF catalogs are backed up after COPY by UCATOPTIONS BACKUP), the user must ensure that the source ICF catalogs are logically at the same point in time as when the volumes were copied.

COPY command syntax

COPY

Required Keywords:

```
{ JOURNAL-DSN( data set name ) | JOURNAL-DDN( ddname ) }
USERCATALOGS( sourcecat1 [ ( volser1 ) ] targetcat1 ...[ , sourcecatn
  [ ( volsern ) ] targetcatn ] ) |
USERCATALOGS-DDN( ddname ) |
USERCATALOGS-NOBACKUP( sourcecat1 [ ( volser1 ) ] targetcat1 ...[ ,
  sourcecatn [ ( volsern ) ] targetcatn ] ) |
USERCATALOGS-BACKUPFIRST( sourcecat1 targetcat1 ...[ , sourcecatn
  targetcatn ] ) |
NOUSERCATALOGS
```

Required only if DATA-MOVER(PGM(ADRDSSU)) is specified (FlashCopy or SnapShot) or DATA-MOVER(PGM(EMCSNAP)) is specified (EMC

TimeFinder/Clone volume snap):

```
{ [ FROM-STORAGEGROUP( storgrp1 | storgrpmask1 ...[ , storgrpn | storgrpmaskn ] ) ]
  [ FROM-USER-STORAGEGROUP( storgrp1 | storgrpmask1 ...
    [ , storgrpn | storgrpmaskn ] ) ]
  [ FROM-VOLSER( volser1 | volmask1 ...[ , volsern | volmaskn ] )
    | FROM-VOLSER-DDN( ddname ) ] }
{ [ TO-STORAGEGROUP( storgrp1 | storgrpmask1 ...[ , storgrpn | storgrpmaskn ] ) ]
  [ TO-USER-STORAGEGROUP( storgrp1 | storgrpmask1 ...
    [ , storgrpn | storgrpmaskn ] ) ]
  [ TO-VOLSER( volser1 | volmask1 ...[ , volsern | volmaskn ] )
    | TO-VOLSER-DDN( ddname ) ] }
```

Required only if DATA-MOVER(PGM(NONE)) is specified:

```
{ VOLPAIRS( sourcevolser1 targetvolser1 ...[ , sourcevolsern targetvolsern ] ) |
  VOLPAIRS-DDN( ddname ) |
VOLPAIRSDEVN( sourcevolser1 targetvolser1 devn1 ... [ , sourcevolsern
  targetvolsern devnn ] ) | VOLPAIRSDEVN-DDN( ddname ) |
VOLPAIRSDEVN-NOCLIP ( sourcevolser1 targetvolser1 devn1
  , sourcevolsern targetvolsern devnn [ , ... ] ) |
VOLPAIRSDEVN-NOCLIP-DDN ( ddname ) }
```

Required only if FROM-USER-STORAGEGROUP or TO-USER-STORAGEGROUP is specified:

```
USERSGDEFS-DDN( ddname )
USERSGDEFS-OFFSETS( INCLXCL(nn), SGNAM(nn), VOLSER(nn) )
```

Required only if USERCATALOGS or USERCATALOGS-NOBACKUP is specified:

```
CATWORK-DSN( mask )
```

Optional Keywords:

```
CATWORK-ATTR( UNIT( SYSALLDA ) SPACE( 10 10 ) CYLINDERS )
DATA-MOVER( [ PGM( ADRDSSU | EMCSNAP | NONE ) ]
  [ , BACKGROUNDCOPY( NO | YES ) ]
  [ , CHECKONLINEPATHSTATUS( NO | YES ) ]
  [ , CHECKVTOC ]
```

```

[ , CONSISTENT( NO | YES ) ]
[ , DIFFERENTIAL( NO | YES ) ]
[ , DSSPARM( ABEND=nnn [ , AMSGCNT=nnnn ] | SDUMP=nnn [ , SMSGCNT=nnnn ] ) ]
[ , COPYCMDLIMIT( nnn | 24 ) ]
[ , FASTREP( PREF | REQ | NONE ) ]
[ , FCNOCOPY ]
[ , FCSETGTOK ]
[ , FCTOPPRCPRIARY [ (PRESMIRREQ | PRESMIRPREF | PRESMIRNONE ) ]
[ , INCREMENTAL( NO | YES ) ]
[ , NOCONCURRENT ]
[ , MAXIMUM-SUBTASKS(number1,number2) ] )
EXCLUDE-FROM-VOLSER( volser1 | volmask1 ...[ volsern | volmaskn ] )
EXCLUDE-TO-VOLSER( volser1 | volmask1 ...[ volsern | volmaskn ] )
SIMULATE
SOURCESONLINE( Y | N )
TARGETSONLINE( Y | N )
TARGETSUONLINE( Y | N )
TARGET-VOLS-SHOULD-BE-EMPTY( Y | N )

```

About source and target catalog names

It is valid for source and target catalog names to be the same. If source and target catalog names are the same, the target catalog will be populated. See the RECATALOG option of the "RENAME" on page 433 and the "BCSCLEAN" on page 345 command.

For example, when:

- Source volumes contain data sets named A1... and B1...:
- Alias A1 points to UCATA1 and alias B1 points to UCATB1
- Alias A2 points to UCATA2 and alias B2 points to UCATB2
- Rename masks (see "RENAME" on page 433): A1.** A2.** B1.** B2.**
USERCATALOGS(UCATA1 UCATA2 UCATB1 UCATB2)

Then:

1. Any source volume data sets matching A1.** are renamed to A2.**.
2. A2.** data sets are cataloged to UCATA2 because source data sets A1.** were found to be cataloged in UCATA1.
3. Any source volume data sets matching B1.** are renamed to B2.**.
4. B2.** data sets are cataloged to UCATB2 because source data sets B1.** were found to be cataloged in UCATB1.

Note: Catalog data must be captured that reflects the status of the source volume data sets at the time of the copies. Some data needed to catalog renamed data sets, such as catalog PATH entries and GDG base records, exist in the catalog only. DB2 Cloning Tool does not attempt to ascertain involved catalogs automatically, due to the time this would take and the consequent delay of source volume access.

FlashCopy for backups

When DB2 Cloning Tool executes the COPY command, it uses the FlashCopy default of 'background COPY' from the source to target volume after the logical completion occurs.

If you want to copy and rename the target volume data sets to be used as input to a backup, FlashCopy's 'NO background COPY' (FCNOCOPY) should be used. If you take a point-in-time copy using FlashCopy, and the target volume is only needed for a short time, such as for input to a backup, copying tracks that haven't

changed would be a waste of resources compared to just using the pointer to the corresponding source volume data set. This concept is called 'NO background COPY'. The 'before image' of tracks that change on the source volume must in fact be created on the target. But, assuming that a DFSMSdss or FDR DUMP of the target volume is started just after the FlashCopy initiation is complete, most likely not many source volume tracks will change in the time it takes the backup to finish.

When the backup is finished however, this NO background COPY process should be stopped to prevent changed data from continually being copied to the target volume.

If you use DFSMSdss to back up the FlashCopy target, a DFSMSdss DUMP parameter, FlashCopy Withdraw (FCWITHDRAW), can be used on the DFSMSdss DUMP to tell DSS to withdraw the FlashCopy relationship when the backup is complete.

If you use FDR as your dump tool, DB2 Cloning Tool provides the COPYCHECK WITHDRAW command to withdraw the FlashCopy relationship. For more information, see the topic "COPYCHECK" on page 371.

Note: If either the source or target volumes are extensively updated during the backup, this option should not be used because excessive overhead will occur when copying changed tracks to the target before allowing the update to occur.

COPY command and keyword definitions

Required keywords are described first, followed by optional keywords.

COPY The COPY command initiates volume copies and in parallel, backs up the source ICF catalogs that point to data sets on the source volumes being cloned.

- Required: Yes
- Restrictions: None

CATWORK-DSN (*mask*)

Specifies a mask used to derive data sets names for catalog backup data sets dynamically allocated during the COPY step.

The mask must include an asterisk (*) as one qualifier. DB2 Cloning Tool will create data sets by substituting two eight-byte qualifiers in place of the provided asterisk. Hence, because 17 bytes (8+the dot+8) of the name will be generated, the user is responsible for the resolved names not exceeding 44 characters (e.g., CATWORK-DSN(CKZ.CATWORK.*) will cause data sets to be created such as: CKZ.CATWORK.UCATBKUP.BKP00001

The asterisk in the mask does not need to be the lowest level qualifier. (e.g., CATWORK(CKZ.CWORK*.DATA)

- Default: None
- Required: Required only if USERCATALOGS or USERCATALOGS-NOBACKUP is specified.
- Restrictions: None
- Short form: CWDSN

JOURNAL-DSN (*data set name*) or JOURNAL-DDN (*ddname*)

Optional syntax is: JRNL-DSN or JRNL-DDN

This parameter supplies either the data set name of the DB2 Cloning Tool journal file, or a DD name assumed via the JCL to point at a journal data set.

If multiple DB2 Cloning Tool setups are used for different volume groups, DO NOT use the same journal data set. Each DB2 Cloning Tool 'application' needs a different journal data set.

The journal is used to pass information between DB2 Cloning Tool steps (e.g., from the COPY step to the RENAME step). Therefore, as noted in the JCL comments, it must be cataloged in the COPY step when first created, and referenced as OLD in subsequent steps.

Do not delete the data set in the last step, in case restarts or reruns need journal information. It is preferable to delete and replace the data set only at the fresh start of a COPY.

- Default: None
- Required: Yes
- Restrictions: None
- Short form: JRNL-DSN JRNL-DDN

USERCATALOGS (*sourcecat1* [(*volser1*)] *targetcat1* ... [, *sourcecatn* [(*volsern*)] *targetcatn*])

This parameter specifies source ICF catalogs that data sets from source (from) volumes are cataloged in, and the corresponding target catalog that renamed volume data sets are to be cataloged in.

ICF catalog names are specified in pairs of source and target. For each renamed data set, the USERCATALOGS list is searched for the catalog the source volume data set was cataloged in. The renamed data set is cataloged in the corresponding target catalog.

The source ICF catalogs specified must include all ICF catalogs that any data set being copied and renamed could be cataloged in.

Target ICF catalogs used to catalog the renamed data sets cannot reside on a target volume during the timeframe from the volume copy through the completion of the RENAME step. If desired, you can move the target ICF catalog(s) from the target volume prior to the volume copy, and move the target ICF catalog(s) back to the target volume after the RENAME has completed.

If you specify the VOLSER with the source ICF catalog name, the source catalog will be backed up from its copy on the corresponding target volume. The VOLSER specified is the source volume where the source ICF catalog resides. The volume where the source ICF catalog resides must be included as one of the source volumes being copied. The VOLSER of the source ICF catalog is specified because when DB2 Cloning Tool does the volume pairing, there is no guarantee the source volume will be paired to the same target volume for every cloning. DB2 Cloning Tool uses the volume pairing information to determine the target volume VOLSER that corresponds with the specified source volume VOLSER.

- Default: None
- Required: Yes
- Restrictions: Mutually exclusive with NOUSERCATALOGS, USERCATALOGS-BACKUPFIRST, USERCATALOGS-NOBACKUP, and USERCATALOGS-DDN.
- Short form: UCATS

USERCATALOGS-NOBACKUP (*sourcecat1* [(*volser1*)] *targetcat1* ... [, *sourcecatn* [(*volsern*)] *targetcatn*]

This parameter specifies source catalogs that data sets from source (from) volumes are cataloged in, and the corresponding target catalog that renamed volume data sets are to be cataloged in.

ICF catalog names are specified in pairs of source and target. For each renamed data set, the USERCATALOGS list is searched for the ICF catalog the source volume data set was cataloged in. The renamed data set is cataloged in the corresponding target catalog.

The source ICF catalogs specified must include all ICF catalogs that any data set being copied and renamed could be cataloged in.

Target ICF catalogs used to catalog the renamed data sets cannot reside on a target volume during the timeframe from the volume copy through the completion of the RENAME step. If desired, you can move the target ICF catalog(s) from the target volume prior to the volume copy, and move the target ICF catalog(s) back to the target volume after the RENAME has completed.

If you specify the VOLSER with the source ICF catalog name, the source catalog will be backed up from its copy on the corresponding target volume. The VOLSER specified is the source volume where the source ICF catalog resides. The volume where the source ICF catalog resides must be included as one of the source volumes being copied. The VOLSER of the source ICF catalog is specified because when DB2 Cloning Tool does the volume pairing, there is no guarantee the source volume will be paired to the same target volume for every cloning. DB2 Cloning Tool uses the volume pairing information to determine the target volume VOLSER that corresponds with the specified source volume VOLSER.

The source catalogs will not be backed up by COPY. The source catalogs will be backed up when UCATOPTIONS BACKUP is run. The backup of the source catalogs with UCATOPTIONS BACKUP must happen prior to running the RENAME command. If VOLSER is specified for the source catalog, UCATOPTIONS BACKUP will make its backup from the copies of the source catalogs on the target volumes.

Note: The ICF catalog backup can be postponed until after COPY by using the USERCATALOGS-NOBACKUP keyword. However, if you choose to postpone the backup, the source ICF catalog information being backed up needs to be in synchronization with the contents of the copied volumes. If USERCATALOGS-NOBACKUP is used (so the source ICF catalogs are backed up after COPY by UCATOPTIONS BACKUP), the user must ensure that the source ICF catalogs are logically at the same point in time as when the volumes were copied.

If VOLPAIRSDEVN-NOCLIP or VOLPAIRSDEVN-NOCLIP-DDN are used with USERCATALOGS-NOBACKUP, VOLOPTIONS OFFLINECLIP must be run before UCATOPTIONS BACKUP which must be run before the RENAME command.

- Default: None
- Required: Yes
- Restrictions: Mutually exclusive with NOUSERCATALOGS, USERCATALOGS, USERCATALOGS-DDN, and USERCATALOGS-BACKUPFIRST.
- Short form: UCATSNB

USERCATALOGS-BACKUPFIRST (*sourcecat1 targetcat1 ... [, sourcecatn targetcatn]*)
This parameter specifies source ICF catalogs that data sets from source (from) volumes are cataloged in, and the corresponding target catalog that renamed volume data sets are to be cataloged in.

ICF catalog names are specified in pairs of source and target. For each renamed data set, the USERCATALOGS list is searched for the catalog the source volume data set was cataloged in. The renamed data set is cataloged in the corresponding target catalog.

The source ICF catalogs specified must include all ICF catalogs that any data set being copied and renamed could be cataloged in.

Target ICF catalogs used to catalog the renamed data sets cannot reside on a target volume during the timeframe from the volume copy through the completion of the RENAME step. If desired, you can move the target ICF catalog(s) from the target volume prior to the volume copy, and move the target ICF catalog(s) back to the target volume after the RENAME has completed.

The source ICF catalogs will be backed up before the target volumes are processed by the COPY command.

- Default: None
- Required: Yes
- Restrictions: Mutually exclusive with NOUSERCATALOGS, USERCATALOGS, USERCATALOGS-DDN, and USERCATALOGS-NOBACKUP.
- Short form: UCATSBF

USERCATALOGS-DDN(*ddname*)

This parameter specifies the DD name which points to a file containing user catalog pairs. The pairs are the same format as in the USERCATALOGS keyword.

USERCATALOGS-DDN must have an LRECL of 80, and must not contain sequence numbers in columns 73 - 80.

- Default: None
- Required: Yes
- Restrictions: Mutually exclusive with USERCATALOGS, NOUSERCATALOGS, USERCATALOGS-NOBACKUP, and USERCATALOGS-BACKUPFIRST.
- Short form: UCATSDDN

NOUSERCATALOGS

Specifies that the COPY will not include the backing up of any ICF catalogs. NOUSERCATALOGS is intended to be used for the following cases:

- When creating a copy of the source volumes where the target volumes are strictly for backup purposes (there is no intent of renaming the data sets on the target volumes).
- When performing a volume cloning using interim volumes. Refer to "Cloning scenarios" on page 535 for more information about this type of cloning.

RENAME is not possible when NOUSERCATALOGS is used.

- Default: None
- Required: No

- Restrictions: Mutually exclusive with USERCATALOGS, USERCATALOGS-DDN, USERCATALOGS-NOBACKUP and USERCATALOGS-BACKUPFIRST.
- Short form: NOUCATS

FROM-STORAGEGROUP (*storgrp1* | *storgrpmask1* ...[, *storgrp*n** | *storgrpmask*n**])

Specifies the input volumes to be copied from one or more SMS storage group definitions or storage groups matching a mask. All volumes from the storage groups specified will be copied, except any volumes excluded via the EXCLUDE-FROM-VOLSER parameter.

An input (source) storage group may only be specified or referred to once per execution. DB2 Cloning Tool will match source volumes with target volumes, in keeping with the requirement that each pair is in the same ESS subsystem and LSS (if ESS FlashCopy V1, same partition if RVA), same track format, and same volume size.

- Default: None
- Required: Required only if DATA-MOVER(PGM(ADRDSSU)) is specified (FlashCopy or SnapShot).
- Restrictions: Valid only with DATA-MOVER(PGM(ADRDSSU))
- Short form: FRS

FROM-USER-STORAGEGROUP (*storgrp1* | *storgrpmask1* ...[, *storgrp*n** | *storgrpmask*n**])

or FROMUSERSTORAGEGROUP

Specifies the input volumes to be copied from one or more user storage group definitions or storage groups matching a mask. All online volumes from the storage groups specified will be copied, except any volumes excluded via the EXCLUDE-FROM-VOLSER parameter.

An input (source) storage group may only be specified or referred to once per execution.

DB2 Cloning Tool will match source volumes with target volumes, in keeping with the requirement that each pair is in the same ESS subsystem and LSS (if ESS FlashCopy V1, same partition if RVA), same track format, and same volume size.

The storage group definitions that are used will be read from the DD statement defined by the USERSGDEFS-DDN keyword.

- Default: None
- Required: Required only if DATA-MOVER(PGM(ADRDSSU)) is specified (FlashCopy or SnapShot).
- Valid only with DATA-MOVER(PGM(ADRDSSU))
- Short form: FRUS

FROM-VOLSER (*volser1* | *volmask1* ...[, *volser*n** | *volmask*n**])

Optional syntax: FROMVOLSER

Specifies the input volumes to be copied, either by discrete volume serial numbers or volume serial masks (e.g., TSO*). See the EXCLUDE-FROM-VOLSER keyword to exclude VOLSERs from a list.

DB2 Cloning Tool will match source volumes with target volumes, in keeping with the requirement that each pair is in the same ESS subsystem and LSS (if ESS FlashCopy V1, same partition if RVA), same track format, and same volume size.

If TO-VOLSER is used in conjunction with FROM-VOLSER, volumes will pair one-for-one (i.e., 1st FROM VOLSER to 1st TO VOLSER, etc.), if the pairing requirements permit this to happen.

- Default: None
- Required: Required only if DATA-MOVER(PGM(ADRDSSU)) is specified (FlashCopy or SnapShot).
- Restrictions: If DATA-MOVER(PGM(NONE)) is used, a volume mask cannot be specified.
- Short form: FRV

FROM-VOLSER-DDN(*ddname*)

Optional syntax: FROMVOLSERDDN

This parameter specifies the DD name that points to a file containing FROM VOLSER volumes. The volumes are the same format as in the FROM-VOLSER keyword.

FROM-VOLSER-DDN must have an LRECL of 80, and must not contain sequence numbers in columns 73 - 80.

- Default: None
- Required: Required only if DATA-MOVER(PGM(ADRDSSU)) is specified (FlashCopy or SnapShot).
- Restrictions: If DATA-MOVER(PGM(NONE)) is used, a volume mask cannot be specified.
- Short form: FRVDDN

TO-STORAGEGROUP (*storgrp1* | *storgrpmask1* ...[, *storgrp*n | *storgrpmask*n**])**

Optional syntax: TOSTORAGEGROUP

Specifies that output volumes needed to pair with input volumes are to be selected from one or more SMS storage groups or storage groups matching a mask. All volumes from the storage groups specified are target candidates, except any volumes excluded via the EXCLUDE-TO-VOLSER parameter.

CKZ will match source volumes with target volumes, in keeping with the requirement that each pair is in the same ESS subsystem and LSS (if ESS FlashCopy V1, same partition if RVA), same track format, and same volume size.

If FROM-STORAGEGROUP is used, DB2 Cloning Tool will attempt to match volumes using the same positioned storage group names listed in TO-STORAGEGROUP. If a hardware difference or other issue results in a FROM-STORAGEGROUP volume that does not match its TO-STORAGEGROUP counterpart, the TO-STORAGEGROUP volume is skipped. DB2 Cloning Tool then attempts to match the FROM-STORAGEGROUP volume with the next TO-STORAGEGROUP volume in the list.

- Default: None
- Required: Required only if DATA-MOVER(PGM(ADRDSSU)) is specified (FlashCopy or SnapShot).
- Restrictions: Valid only with DATA-MOVER(PGM(ADRDSSU))
- Short form: TOS

TO-USER-STORAGEGROUP (*storgrp1* | *storgrpmask1* ...[, *storgrp*n | *storgrpmask*n**])** Optional syntax: TOUSERSTORAGEGROUP

Specifies that output volumes needed to pair with input volumes are to be selected from one or more user storage groups or storage groups matching a mask. All online volumes from the storage groups specified are target candidates, except any volumes excluded via the EXCLUDE-TO-VOLSER parameter.

DB2 Cloning Tool will match source volumes with target volumes, in keeping with the requirement that each pair is in the same ESS subsystem and LSS (if ESS FlashCopy V1, same partition if RVA), same track format, and same volume size.

If FROM-USER-STORAGEGROUP is used, DB2 Cloning Tool will attempt to match volumes using the same positioned storage group names listed in TO-USER-STORAGEGROUP. If a hardware difference or other issue results in a FROM-USER-STORAGEGROUP volume that does not match its TO-USER-STORAGEGROUP counterpart, the TO-USER-STORAGEGROUP volume is skipped. DB2 Cloning Tool then attempts to match the FROM-USER-STORAGEGROUP volume with the next TO-USER-STORAGEGROUP volume in the list.

- Default: None
- Required: Required only if DATA-MOVER(PGM(ADRDSSU)) is specified (FlashCopy or SnapShot).
- Restrictions: Valid only with DATA-MOVER(PGM(ADRDSSU))
- Short form: TOUS

TO-VOLSER (*volser1* | *volmask1* ...[, *volsern* | *volmaskn*])

Optional syntax: TOVOLSER

Specifies target volumes to be paired with input volumes.

DB2 Cloning Tool will match source volumes with target volumes, in keeping with the requirement that each pair is in the same ESS subsystem and LSS (if ESS FlashCopy V1, same partition if RVA), same track format, and same volume size. If TO-VOLSER is used in conjunction with FROM-VOLSER, volumes will pair one-for-one (i.e., 1st FROM VOLSER to 1st TO VOLSER etc.), if the pairing requirements permit this to happen.

- Default: None
- Required: Required only if DATA-MOVER(PGM(ADRDSSU)) is specified (FlashCopy or SnapShot).
- Restrictions: If DATA-MOVER(PGM(NONE)) is used, a volume mask cannot be specified.
- Short form: TOV

TO-VOLSER-DDN(*ddname*)

Optional syntax: TOVOLSERDDN

This parameter specifies the DD name that points to a file containing to volser volumes. The volumes are the same format as in the TO-VOLSER keyword.

TO-VOLSER-DDN must have an LRECL of 80, and must not contain sequence numbers in columns 73-80.

- Default: None
- Required: Required only if DATA-MOVER(PGM(ADRDSSU)) is specified (FlashCopy or SnapShot).
- Restrictions: If DATA-MOVER(PGM(NONE)) is used, a volume mask cannot be specified. Mutually exclusive with TO-VOLSER.

- Short form: TOVDDN

USERSGDEFS-DDN (*ddname*)

Specifies a DD name which points to a file containing the user storage group definitions that will be used by the FROM-USER-STORAGEGROUP and TO-USER-STORAGEGROUP keywords. The USERSGDEFS-OFFSET keyword is used to specify the offsets of the fields within the records.

The file must have an LRECL of 80. A comment record is denoted by a blank or * in column 1 or a /* in columns 1 and 2.

- Default: None
- Required: Required only if FROM-USER-STORAGEGROUP or TO-USER-STORAGEGROUP are specified.
- Restrictions: Used only if FROM-USER-STORAGEGROUP or TO-USER-STORAGEGROUP are specified.

USERSGDEFS-OFFSETS (*INCLEXCL(mm)*, *SGNAME(mm)*, *VOLSER(mm)*)

Specifies the offsets in the record that the fields will be found. These fields are used for the user storage group definitions that will be used by the FROM-USER-STORAGEGROUP and TO-USER-STORAGEGROUP keywords.

The offsets specified can be from 1 to 72 inclusive, with 1 being the first character position in the record.

INCLEXCL specifies the offset where the 1-character include/exclude indicator can be found. A value of blank or I indicates an include definition. A value of E or X indicates an exclude definition.

SGNAME specifies the offset where the 8-character storage group name can be found.

VOLSER specifies the offset where the 6-character volser or volser mask can be found.

For example: Using USERSGDEFS-OFFSETS(*VOLSER(1)*, *INCLEXCL(8)*, *SGNAME(10)*) and USERSGDEFS-DDN file containing the records:

```
SRC00*  SGSRC01
SRC006 E SGSRC01
TGT01* I SGTGT01
TGT02* I SGTGT01
TGT015 X SGTGT01
TGT025 X SGTGT01
```

FROM-USER-STORAGEGROUPS(*SGSRC01*) would resolve to all online DASD volumes that match the mask SRC00* except for SRC006 which would be excluded.

TO-USER-STORAGEGROUPS(*SGTGT01*) would resolve to all online DASD volumes that match the masks TGT01* and TGT02* except that TGT015 and TGT025 would be excluded.

- Default: None
- Required: Required only if FROM-USER-STORAGEGROUP or TO-USER-STORAGEGROUP are specified.
- Restrictions: Used only if FROM-USER-STORAGEGROUP or TO-USER-STORAGEGROUP are specified.

VOLPAIRS (*sourcevolser1 targetvolser1 ... [, sourcevolsern targetvolsern]*)

Optional syntax:

- VOLPAIRS-DDN(*ddname*) |
- VOLPAIRSDEVN (*sourcevolser1 targetvolser1 devn1 ... [, sourcevolsern targetvolsern devnn]*)
- VOLPAIRSDEVN-DDN(*ddname*) |
- VOLPAIRSDEVN-NOCLIP (*sourcevolser1 targetvolser1 devn1 ... [, sourcevolsern targetvolsern devnn]*)
- VOLPAIRSDEVN-NOCLIP-DDN (*ddname*)

VOLPAIRS specifies volume pairs where the target volumes have been created by the user before the DB2 Cloning Tool COPY command is executed. Target volumes must have the desired internal VOLSER (not the VOLSER of the source volume) and be online. Discrete VOLSERS only, no masking allowed. Short form is VP.

VOLPAIRS-DDN specifies a DD name which points to a file containing the volume pairs. The pairs are the same format as in the VOLPAIRS keyword. VOLPAIRS-DDN must have an LRECL of 80, and must not contain sequence numbers in columns 73-80.

VOLPAIRSDEVN specifies volume pairs with the device number of the target volume where target volumes have been created by the user before the DB2 Cloning Tool copy command is executed. Target volumes must be offline and the internal VOLSER must match the corresponding source VOLSER specified. DB2 Cloning Tool will re-label the specified device (*devn*) to the corresponding target VOLSER and vary the volume online. Short form is VPD.

VOLPAIRSDEVN-DDN specifies a DD name which points to a file containing the volume pairs. The pairs are the same format as in the VOLPAIRSDEVN keyword. VOLPAIRSDEVN-DDN must have an LRECL of 80, and must not contain sequence numbers in columns 73-80.

VOLPAIRSDEVN-NOCLIP specifies volume pairs with the device number of the target volume where target volumes have been created by the user before the DB2 Cloning Tool COPY command is executed. Target volumes must be offline and the internal VOLSER must match the corresponding source VOLSER specified. COPY will not re-label the specified device (*devn*) or vary the volume online. The VOLOPTIONS OFFLINECLIP command can be used to re-label the specified target devices and vary the volumes online. The re-label and vary online of the target volumes with VOLOPTIONS OFFLINECLIP must happen prior to running the RENAME command. If VOLPAIRSDEVN-NOCLIP or VOLPAIRSDEVN-NOCLIP-DDN are used with USERCATALOGS-NOBACKUP, VOLOPTIONS OFFLINECLIP must be run before UCATOPTIONS BACKUP which must be run before the RENAME command.

VOLPAIRSDEVN-NOCLIP-DDN specifies a *ddname* which points to a file containing the volume pairs. The pairs are the same format as in the VOLPAIRSDEVN-NOCLIP keyword. VOLPAIRSDEVN-NOCLIP-DDN must have an LRECL of 80, and must not contain sequence numbers in columns 73-80.

- Default: None
- Required: One of the VOLPAIRS keywords is required only if DATA-MOVER(PGM(NONE)) is specified.
- Restrictions: Valid only with DATA-MOVER(PGM(NONE)).
- Short form: FRV

CATWORK-ATTR (*catalog backup allocation attributes*)

Specifies allocation attributes used when catalog backup data sets are dynamically allocated. Unless unusual attributes are desired for a specific DB2 Cloning Tool COPY, these attributes can be globally specified in the HLQ?.SCKZPARM. CKZINI member, :COPY_OPTIONS section, CATWORK-ATTR token. Allocation attributes are specified in TSO allocate syntax (e.g., UNIT(SYSDA) SPACE(1 1) TRACKS, etc.).

The attributes that can be specified include:

- DATACLAS(data class name)
- MGMTCLAS(management class name)
- SPACE(quantity increment)
- STORCLAS(storage class name)
- TRACKS/CYLINDERS UNIT(unit)
- VOLUME(serial)

If an initial attempt running COPY fails because a catalog backup data set exceeds extents, increase the allocation and run again. Once successful, examine the space actually used and decrease if desired. To accommodate a future increase in the size of catalogs, leave the allocation with room to spare.

- Default: UNIT(SYSALLDA) SPACE(10 10) CYLINDERS
- Required: No.
- Restrictions: None
- Short form: CWATTR

DATA-MOVER ([PGM(ADRDSSU | EMCSNAP | NONE)]

For *ADRDSSU*:

```
[ , CHECKVTOC ]
[ , CONSISTENT( NO | YES ) ]
[ , COPYCMDLIMIT( nnn | 24 ) ]
[ , DSSPARM( ABEND=nnn [ , AMSGCNT=nnnn ] | SDUMP=nnn [ ,
SMSGCNT=nnnn ] ) ]
[ , FASTREP( PREF | REQ | NONE ) ]
[ , FCNOCOPY ]
[ , FCSETGTOK ]
[ , FCTOPPRCPRIARY [ (PRESMIRREQ | PRESMIRPREF |
PRESMIRNONE ) ]
[ , INCREMENTAL( NO | YES ) ]
[ , NOCONCURRENT ] )
```

For *EMCSNAP*:

```
[ , BACKGROUNDCOPY( NO | YES ) ]
[ , CHECKONLINEPATHSTATUS( NO | YES ) ]
[ , CONSISTENT( NO | YES ) ]
[ , DIFFERENTIAL( NO | YES ) ]
[ , MAXIMUM-SUBTASKS( number1, number2 ) ] )
```

In all cases, specifies the program to be used to initiate copies and copy options.

ADRDSSU specifies that COPY is to initiate FlashCopy or SnapShot 'under the covers' via execution of DSS. It is the default and therefore the PGM(ADRDSSU) keyword can be omitted. DB2 Cloning Tool invokes DSS 'under the covers' to initiate volume copies with the COPY FULL option.

Users should be acquainted with the DSS rules governing copy full operations (IBM publication Advanced Copy Services, Combining Storage Control Copy Operations).

For a COPY FULL operation, DSS determines by the device types of the volumes to be copied what copy mechanism is to be used. DB2 Cloning Tool pairs volumes, so that FlashCopy or SnapShot should be used, and confirms at the time of the COPY step that conflicting relationships do not exist. However, because of the small window between DB2 Cloning Tool volume pairing and relationship validation, DSS errors and associated messages may need to be examined.

DB2 Cloning Tool COPY command supplies the ADMINISTRATOR operand when invoking DFSMSdss. To avoid WTORs, ADRDSSU ADMINISTRATOR is used to gain permission to overlay the target volume VTOCIX and/or VVDS during the COPY process.

Because the ADMINISTRATOR operand is generated, the userid running COPY must have READ access to FACILITY class profile STGADMIN.ADR.STGADMIN.COPY.

EMCSNAP specifies that COPY is to initiate EMC SNAP 'under the covers' via EMC TimeFinder/Clone Mainframe Snap Facility's volume level support. DB2 Cloning Tool invokes EMC TimeFinder/Clone 'under the covers' to initiate volume copies using SNAP VOLUME commands. Users should be acquainted with the EMC rules governing SNAP VOLUME operations (EMC publication TimeFinder/Clone Mainframe SNAP Facility Product Guide).

DB2 Cloning Tool pairs volumes, so that EMC TimeFinder/Clone volume snap should be used, and confirms at the time of the COPY step that conflicting relationships do not exist. However, because of the small window between DB2 Cloning Tool volume pairing and relationship validation, EMC TimeFinder/Clone volume snap errors and associated messages may need to be examined.

NONE specifies that no DATA-MOVER is to be invoked by COPY. NONE infers that volume copies have been created by the user prior to the execution of DB2 Cloning Tool. When NONE is specified, COPY still captures necessary catalog information, and if VOLPAIRSDEVN or VOLPAIRSDEVN-DDN is specified, clips and varies target volumes online.

The following parameters are for ADRDSSU:

CHECKVTOC (DSS parameter): Specifies that a VTOC analysis of the source volume be performed during copy processing.

CONSISTENT (DB2 Cloning Tool parameter): Indicates to use a FlashCopy Consistency Group (YES) or not (NO). The default is NO. This does not apply when FASTREP(NONE or PREF) is also specified. To establish a Consistency Group, a FCCGFREEZE parameter is added to each COPY statement passed to ADRDSSU. This will cause I/O activity to the FlashCopy source volumes to be held (frozen). Once all the COPY statements have been processed by ADRDSSU, a CGCREATE command is passed to ADRDSSU. This will cause I/O activity to be resumed on the "frozen" FlashCopy source volumes. The number of volumes being copied will affect the amount of time the I/O activity to the source volumes will be held. If this time exceeds the smallest Consistency Group timer value defined for a LSS, the target volumes will not be consistent.

COPYCMDLIMIT(*nnn* | 24) (DB2 Cloning Tool parameter): Specifies the maximum COPY FULL commands built by DB2 Cloning Tool for each DSS execution. 24 is the default if omitted. If the number of volumes to be copied exceeds the COPYCMDLIMIT, DB2 Cloning Tool will invoke DSS as many times as necessary. Adjusting this value may affect the performance of ESS copy initiations.

DSSPARM (DSS parameter): This parameter can be used to pass execution parameters to ADRDSSU. This parameter is intended for diagnostic purposes and can be used to gather more information when ADRDSSU has problems. The parameter values should be: ABEND=*nnn*[,*AMSGCNT*=*nnnn*] or SDUMP=*nnn*[,*SMSGCNT*=*nnnn*]. These parameters are described in the DFSMS Storage Administration Reference manual in the section “How to Control DFSMSdss through PARM Information in the EXEC Statement”.

FASTREP (DSS parameter): Indicates if fast replication is preferred (PREF), required (REQ), or not required (NONE). DB2 Cloning Tool will set up the source/target pairs for a fast replication if PREF or REQ is specified. DB2 Cloning Tool will allow a ‘normal’ copy if NONE is specified. If the level of ADRDSSU indicates it supports this keyword, the keyword will be passed to ADRDSSU.

FCNOCOPY: (DSS parameter) Indicates that no background copy should be done for the volume pair. This applies to ESS devices only.

Note: When using FCNOCOPY, you must terminate the FlashCopy source and target relationship at the completion of the backup. Either use the DFSMSdss DUMP parameter FCWITHDRAW, or use COPYCHECK WITHDRAW. Following the withdraw, some of the tracks on the volume may contain data from the source volume, while other tracks may contain residual data that was on the target volume before the copy. This situation can cause problems when trying to access the target volume if the VTOC locations of the source and target volumes were different before the copy.

FCSETGTOK (DSS parameter): Indicates that a FlashCopy target volume can also be a space efficient volume. This does not apply when FASTREP(NONE) is also specified. The physical background copy option is not permitted for space efficient FlashCopy. Because of this, the FCNOCOPY keyword must also be specified with the FCSETGTOK keyword.

FCTOPPRCPRIMARY [(PRESMIRREQ | PRESMIRPREF | PRESMIRNONE)] (DSS parameter): Indicates that a FlashCopy target volume can also be a PPRC primary volume. This applies to ESS devices only. This does not apply when FASTREP(NONE) is also specified.

IBM Remote Pair FlashCopy (also known as Preserve Mirror) can be specified by including one of the optional keywords. Preserve Mirror mirrors the FlashCopy command that is issued at the local site to the remote site. This allows FlashCopy operations to occur to PPRC primary volumes without affecting the PPRC duplex state. IBM Remote Pair FlashCopy must be installed in the storage controller along with the corresponding software support in z/OS. In addition, both the source and target volumes being PPRC primary volumes and in the same storage controller and their corresponding PPRC secondary volumes being in the same storage controller.

Specify one of the following use this functionality:

- PRESMIRREQ (short form: PMR): Require the use of Preserve Mirror. If a Preserve Mirror operation cannot be accomplished, the FlashCopy operation will not be completed.
- PRESMIRPREF (short form: PMP): Prefer the use of Preserve Mirror. If a Preserve Mirror operation cannot be accomplished, the FlashCopy operation is still performed.
- PRESMIRNONE (short form: PMN): Do not use Preserve Mirror.

When any of these options is specified, they will be used as part of the source to target volume pairing criteria. The specified option will also be passed to ADRDSSU as part of the generated copy commands. For additional information about Preserve Mirror, refer to the documentation for your version of z/OS.

INCREMENTAL (DB2 Cloning Tool parameter): Indicates that a full volume Incremental FlashCopy relationship is to be established (YES) or not (NO). The default is NO. This does not apply when FASTREP(NONE or PREF) is also specified. Volumes will be paired using an existing Incremental FlashCopy relationship if possible. INCREMENTAL(YES) cannot be used with FCNOCOPY.

NOCONCURRENT (DSS parameter): Specifies that the CONCURRENT option will not be supplied to ADRDSSU. This will prevent ADRDSSU from using Concurrent Copy when doing the volume copies.

The following parameters are for EMCSNAP:

BACKGROUNDCOPY: Indicates if background copy should be done (YES), or not (NO), for the volume pair. The default is YES.

CHECKONLINEPATHSTATUS: Indicates if a check that paths from other CPUs to the target devices are offline before performing a volume snap (YES) or not (NO). The default is NO.

CONSISTENT: Indicates to use Enginuity Consistency Assist (ECA) for consistent SNAP VOLUME operations (YES) or not (NO). The default is NO.

DIFFERENTIAL: Indicates to use the Enginuity Differential Snap feature for SNAP VOLUME operations (YES) or not (NO). The default is NO.

MAXIMUM-SUBTASKS (number1, number2): Sets an absolute maximum number of subtasks that can be attached and used. SNAP automatically limits the number of subtasks based on the requests specified and the low and high region available. SNAP never exceeds the limits specified in this parameter. *number1* is the limit of the number of individual requests that can be processed simultaneously. The minimum value you can specify is two (2). The maximum value you can specify is 9999. *number2* is the limit of the number of individual activities that can be performed within a single request, typically as the result of wildcarding. The minimum value you can specify is two (2). The maximum value you can specify is 9999. If not specified, the EMC system defaults are used.

- Default: For PGM, the default is ADRDSSU. For COPYCMDLIMIT, the default is 24. For FASTREP, the default is REQ. For INCREMENTAL, the default is NO. For BACKGROUNDCOPY, the default is YES. For CHECKONLINEPATHSTATUS, the default is NO. For CONSISTENT, the default is NO. For DIFFERENTIAL, the default is NO.
- Required: No.

- Restrictions: INCREMENTAL(YES) is mutually exclusive with FCNOCOPY.
- Short form(s): DM, CHECK, CCL, FR, FCNC, FCTOPPRCP, NOCC

EXCLUDE-FROM-VOLSER (*volser1* | *volmask1* ...[*volsern* | *volmaskn*])

Optional syntax: EXCLUDEFROMVOLSER

Specifies volumes or volumes matching a mask, to be excluded from being specified for either the FROM-VOLSER parameter, or the FROM-STORAGEGROUP parameter.

- Default: None
- Required: No
- Restrictions: Not valid with VOLPAIRS, VOLPAIRSDEVN, VOLPAIRSDEVN-DDN, VOLPAIRSDEVN-NOCLIP, or VOLPAIRSDEVN-NOCLIP-DDN.
- Short form: EXCFRV

EXCLUDE-TO-VOLSER (*volser1* | *volmask1* ...[*volsern* | *volmaskn*])

Optional syntax: EXCLUDETOVOLSER

Specifies volumes or volumes matching a mask, to be excluded (not selected as targets) from being specified for either the TO-VOLSER parameter or the TO-STORAGEGROUP parameter.

If the 'eliminated' target volumes cause there to be more source volumes than targets, the COPY will fail.

If the 'eliminated' target volumes still leave at least as many target volumes as source volumes, the pairing will continue as usual.

- Default: None
- Required: No.
- Restrictions: Not valid with VOLPAIRS, VOLPAIRSDEVN, VOLPAIRSDEVN-DDN, VOLPAIRSDEVN-NOCLIP, or VOLPAIRSDEVN-NOCLIP-DDN.
- Short form: EXCTOV

SIMULATE

For DATAMOVER(PGM(ADRDSSU)), SIMULATE specifies that the COPY step is to perform all functions except actual initiation of copies. This option is recommended for new DB2 Cloning Tool setups or when modifications to the setup are made.

SIMULATE will verify syntax, match source to target volumes, display DSS COPY FULL commands (but not initiate the copies), and back up user catalogs. User catalogs are backed up to enable the SIMULATE option of the RENAME step.

Note that in the process of pairing volumes, the logic is exercised to assure that the same ESS subsystem, LSS (FlashCopy V1), and volume size requirements permit all source volumes to be paired with targets. If discrete volumes are specified, and one-for-one pairing is desired, the simulated COPY FULL commands will verify whether pairing requirements were satisfied for a one-for-one pairing.

For DATAMOVER(PGM(NONE)), SIMULATE specifies that the COPY step checks syntax, and backs up the source ICF Catalogs for input to RENAME SIMULATE.

- Default: None.

- Required: No.
- Restrictions: None.
- Short form: SIM

SOURCESONLINE(Y | N)

DB2 Cloning Tool COPY command will expect to locate the source volumes online. If N is specified, DB2 Cloning Tool COPY will not expect to find all the source volumes online.

Note: If N is specified and the source volumes are not online, the source user catalogs must still be available to the DB2 Cloning Tool COPY command.

- Default: Y
- Required: No.
- Restrictions: Valid only with VOLPAIRS, VOLPAIRS-DDN, VOLPAIRSDEVN, VOLPAIRSDEVN-DDN, VOLPAIRSDEVN-NOCLIP, or VOLPAIRSDEVN-NOCLIP-DDN
- Short form: SRCON

TARGETSONLINE(Y | N)

If Y is specified, DB2 Cloning Tool COPY will expect to locate the target volumes online. If N is specified, DB2 Cloning Tool COPY will not expect to find all the target volumes online.

- Default: Y
- Required: No.
- Restrictions: Valid only with VOLPAIRS.
- Short form: TGTON

TARGETSUONLINE(Y | N)

If Y is specified, DB2 Cloning Tool COPY will expect to locate the target user catalogs. It will check that the security product will allow ALTER access to the target user catalogs.

If N is specified, DB2 Cloning Tool COPY will not check for the existence / availability of the target user catalogs. DB2 Cloning Tool RENAME will check that the target catalogs are available and will do the RACF checks.

- Default: Y
- Required: No.
- Short form: TGTUON

TARGET-VOLS-SHOULD-BE-EMPTY(Y | N)

Performs a check during the volume pairing process to ensure the target volumes are empty before issuing FlashCopy or SnapShot. In the event a subsequent RENAME fails and the COPY must be rerun, DB2 Cloning Tool will not clean off the target volumes if 'Y' was specified for this parameter. Either initialize the target volumes or change this keyword to 'N'.

- Default: N
- Required: No.
- Restrictions: Not valid with VOLPAIRS, VOLPAIRSDEVN, VOLPAIRSDEVN-DDN, VOLPAIRSDEVN-NOCLIP, or VOLPAIRSDEVN-NOCLIP-DDN.
- Short form: TVE

Offline Volumes (with DATA-MOVER(PGM(ADDRDSSU)) or PGM(EMCSNAP))

Source volumes:

- Discrete VOLSER specification: Must be found online, or COPY terminates with a return code 8.
- VOLSER mask specification: At least one online volume must be found that matches the mask, or COPY terminates with a return code 8.
- Storage group specification: At least one volume associated with the storage group must be online, or COPY terminates with a return code 8. A warning, return code 4, is generated if at least one volume is not found. Note that DB2 Cloning Tool cannot tell the difference between a non-existent volume and an offline volume.

Target volumes (no matter how they are specified) :

- If enough online volumes can be found to pair with source volumes, COPY will proceed.
- If target volumes are not found, but not needed, COPY will not complain.
- If fewer target volumes are found than needed to pair with source volumes, COPY will terminate with a return code 8.
- Because in some cases DB2 Cloning Tool cannot distinguish between a non-existent volume and an offline volume, messages refer to these volumes as 'not found'.

COPY step JCL example

| Two COPY step JCL examples are included: an example that uses ADDRDSSU to do
| the volume copies, and an example that uses EMC TimeFinder/Clone volume
| snaps to do the volume copies. Sample JCL can be found in the installation library
| SCKZJCL in member CKZCOPY. Both ADDRDSSU and EMC TimeFinder/Clone
| examples are contained in the member.

COPY Step JCL – example that uses ADDRDSSU to do the volume copies

For completeness and to illustrate where JCL parameters and DB2 Cloning Tool control statements must match, the following JCL includes sample DB2 Cloning Tool control statements.

The COPY step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```
1 //????????? JOB , 'CKZ COPY', CLASS=A, MSGCLASS=X
2 //S1 EXEC PGM=IDCAMS
3 //SYSPRINT DD SYSOUT=*
4 //SYSIN DD *
5 1 DEL CKZ.JRNL
6 2 DEL CKZ.WRK.UCATBKUP.*
7 SET MAXCC=0
8 //S1 EXEC PGM=CKZ00010, REGION=8M
9 //STEPLIB DD DSN=HLQ?.SCKZLOAD, DISP=SHR
10 //CKZINI DD DSN=HLQ?.SCKZPARM(CKZINI), DISP=SHR
11 //CKZPRINT DD SYSOUT=*
12 //SYSUDUMP DD SYSOUT=*
13 7 //JOURNAL DD DSN=CKZ.JRNL, RECOG=KS, KEYLEN=64, KEYOFF=0,
14 // DISP=(,CATLG), UNIT=SYSALLDA,
```

```

//          LRECL=600,SPACE=(CYL,(10,10))
//CKZIN   DD  *
COPY                                     -
    DATA-MOVER(                         -
        COPYCMDLIMIT(24)                -
    )                                     -
    FROM-VOLSER(VSRC02)                  -
    TO-VOLSER(VTGT02)                   -
8    USERCATALOGS(                       -
    USERCAT.SRC01 USERCAT.TGT01         -
    USERCAT.SRC02 USERCAT.TGT02         -
    )                                     -
2    CATWORK-DSN(CKZ.WRK.* )           -
7    JOURNAL-DDN(JOURNAL)              -
//*

```

1. Deletion of journal data set in anticipation of allocating new for each execution. Because this data set is used to pass information from one DB2 Cloning Tool step to another, do not delete the journal data set in any steps except the COPY step.
2. Deletion of catalog backup work data sets in anticipation of allocating new for each execution. The DB2 Cloning Tool COPY step backs up each catalog specified by the USERCATALOGS control statement. Output data sets for each catalog backup are dynamically allocated using the CATWORK-DSN(mask) to derive names. Allocation attributes are specified in the CKZINI member of SCKZPARAM:COPY_OPTIONS section, CATWORK-ATTR token. Delete these data sets only in the COPY step.
3. Execution of DB2 Cloning Tool main program.
4. DB2 Cloning Tool SCKZLOAD library must be authorized.
5. DD for CKZINI, SCKZPARAM member. The CKZINI member of the HLQ?.SCKZPARAM library provides variables to the DB2 Cloning Tool programs.
6. DD for DB2 Cloning Tool output.
7. Journal data set used by DB2 Cloning Tool commands. This VSAM data set is used to pass information between DB2 Cloning Tool steps, and to log information derived from the step executions. Each DB2 Cloning Tool 'application' must have a unique journal data set.

Because logging varies based on errors and or warnings discovered in the process, the data set size is difficult to predict. Therefore, test an application using the suggested CYL 10,10 allocation and increase if necessary. After a successful execution, the size may be reduced after examining the total allocation. Allow for changes to the application setup. Number of volumes, number of data sets and number of warnings will affect the required size.

As noted in the IDCAMS step, be sure this data set is not deleted before all DB2 Cloning Tool steps are complete.

In the sample JCL, the control statement JOURNAL-DDN(JOURNAL) specifies that a DD statement with the name JOURNAL is used rather than a dynamically allocated data set name.

8. During the COPY process, DB2 Cloning Tool produces two work files for each catalog pair specified in USERCATALOGS:
 - BKPnnnnn is the backup of the source catalogs.
 - SRTnnnnn is used as the output from the RENAME sort of the BCS records selected by the RENAME-MASKS.

COPY Step JCL – example that uses EMC TimeFinder/Clone volume snaps to do the volume copies

For completeness and to illustrate where JCL parameters and DB2 Cloning Tool control statements must match, the following JCL includes sample DB2 Cloning Tool control statements.

The COPY step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

    //???????? JOB , 'CKZ COPY', CLASS=A, MSGCLASS=X
    //S1 EXEC PGM=IDCAMS
    //SYSPRINT DD SYSOUT=*
    //SYSIN DD *
1   DEL CKZ.JRNL
2   DEL CKZ.WRK.UCATBKUP.*
   SET MAXCC=0
3  //S1 EXEC PGM=CKZ00010, REGION=8M
4  //STEPLIB DD DSN=HLQ?.SCKZLOAD, DISP=SHR
5  //CKZINI DD DSN=HLQ?.SCKZPARM(CKZINI), DISP=SHR
6  //CKZPRINT DD SYSOUT=*
   //SYSUDUMP DD SYSOUT=*
7  //JOURNAL DD DSN=CKZ.JRNL, RECORG=KS, KEYLEN=64, KEYOFF=0,
   // DISP=(,CATLG), UNIT=SYSALLDA,
   // LRECL=600, SPACE=(CYL,(10,10))
   //CKZIN DD *
   COPY
   DATA-MOVER(
8     PGM(EMCSNAP)
   )
   FROM-VOLSER(VSRC02)
   TO-VOLSER(VTGT02)
9   USERCATALOGS(
   USERCAT.SRC01 USERCAT.TGT01
   USERCAT.SRC02 USERCAT.TGT02
   )
2   CATWORK-DSN(CKZ.WRK.* )
7   JOURNAL-DDN(JOURNAL)
   //*

```

1. Deletion of journal data set in anticipation of allocating new for each execution. Because this data set is used to pass information from one DB2 Cloning Tool step to another, do not delete the journal data set in any steps except the COPY step.
2. Deletion of catalog backup work data sets in anticipation of allocating new for each execution. The DB2 Cloning Tool COPY step backs up each catalog specified by the USERCATALOGS control statement. Output data sets for each catalog backup are dynamically allocated using the CATWORK-DSN(mask) to derive names. Allocation attributes are specified in the CKZINI member of SCKZPARM: COPY_OPTIONS section, CATWORK-ATTR token. Delete these data sets only in the COPY step.
3. Execution of DB2 Cloning Tool main program.
4. DB2 Cloning Tool SCKZLOAD library must be authorized.
5. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the DB2 Cloning Tool programs.
6. DD for DB2 Cloning Tool output.
7. Journal data set used by DB2 Cloning Tool commands. This VSAM data set is used to pass information between DB2 Cloning Tool steps, and to log

information derived from the step executions. Each DB2 Cloning Tool 'application' must have a unique journal data set.

Because logging varies based on errors and or warnings discovered in the process, the data set size is difficult to predict. Therefore, test an application using the suggested CYL 10,10 allocation and increase if necessary. After a successful execution, the size may be reduced after examining the total allocation. Allow for changes to the application setup. Number of volumes, number of data sets and number of warnings will affect the required size.

As noted in the IDCAMS step, be sure this data set is not deleted before all DB2 Cloning Tool steps are complete.

In the sample JCL, the control statement JOURNAL-DDN(JOURNAL) specifies that a DD statement with the name JOURNAL is used rather than a dynamically allocated data set name.

8. Requests that the volumes copies be done using EMC TimeFinder/Clone Mainframe Snap Facility's volume level support.
9. During the COPY process, DB2 Cloning Tool produces two work files for each catalog pair specified in USERCATALOGS:
 - BKPnnnnn is the backup of the source catalogs.
 - SRTnnnnn is used as the output from the RENAME sort of the BCS records selected by the RENAME-MASKS.

COPYCHECK

This command is not required. COPYCHECK is provided in case other relationships are to be initiated that require the previously initiated copies to complete, or if COPY must be rerun, and withdrawing previously established copies may save time, rather than waiting for them to finish.

COPYCHECK provides a mechanism to either 'WAIT' for copies to complete, or to 'WITHDRAW' or 'STOPSNAP' (terminate) previously established volume relationships.

- WAIT is intended for situations where other relationships need to be initiated when the DB2 Cloning Tool copies are complete.
- WITHDRAW is intended for reruns of the COPY step without waiting for copies to complete when FlashCopy is used.
- STOPSNAP is intended for reruns of the COPY step without waiting for copies to complete when EMC SNAP is used.

If the copy was established with FCNOCOPY or BACKGROUNDCOPY(NO) (no background copy), following the withdraw, some of the tracks on the volume may contain data from the source volume, while other tracks may contain residual data that was on the target volume before the copy. This situation can cause problems when trying to access the target volume if the VTOC locations of the source and target volumes were different before the copy.

COPYCHECK command syntax

COPYCHECK

Required keywords:

```
{ JOURNAL-DSN( data set name ) | JOURNAL-DDN( ddname ) }  
WAIT( nnn | 10 [ ,RC( rr | 4 ) ] ) | WITHDRAW | STOPSNAP
```

COPYCHECK command and keyword definitions

Required keywords are described first, followed by optional keywords.

COPYCHECK

Optional command to wait for completion of COPY events, or withdraw from them.

- Required: No
- Restrictions: None

JOURNAL-DSN (*data set name*) or JOURNAL-DDN (*ddname*)

This parameter supplies either the data set name of the DB2 Cloning Tool journal file or the DD name of the DD statement in the JCL that points to the DB2 Cloning Tool journal file.

COPYCHECK, whether used to wait for copy completions or to withdraw copy relationships, relies on the volume pairs carried in the journal data set from a previously executed COPY command.

If multiple DB2 Cloning Tool setups are used for different volume groups, DO NOT use the same journal data set. Each DB2 Cloning Tool 'application' needs a different journal data set.

The journal is used to pass information between DB2 Cloning Tool steps (e.g., from the COPY step to the RENAME step). Therefore, as noted in the JCL comments, it must be cataloged in the COPY step when first created, and referenced as OLD in subsequent steps.

Do not delete the data set in the last step, in case restarts or reruns need journal information. It is preferable to delete and replace the data set only at the fresh start of a COPY.

- Default: None
- Required: Yes
- Restrictions: None

WAIT(*mmm* | 10 [,RC(*rr* | 4)])

WAIT specifies that COPYCHECK is to check at 30-second intervals to see if copy relationships have completed for all volume copies initiated in a corresponding COPY step.

mmm specifies the maximum time in minutes that COPYCHECK should continue checking. If the specified time limit expires before all copies are complete, COPYCHECK will terminate with a return code of *rr*.

COPYCHECK will examine the volume copies that are passed via the journal data set from a corresponding COPY step.

- Default: 10,RC(4)
- Required: Yes
- Restrictions: Mutually exclusive with WITHDRAW and STOPSNAP.

WITHDRAW

WITHDRAW specifies that COPYCHECK is to withdraw all copy relationships initiated in a corresponding COPY step.

If COPY must be rerun, WITHDRAW will eliminate the otherwise required wait time for one set of volume copies to complete before copies involving the same volumes can be initiated again.

- Default: None

- Required: Yes
- Restrictions: Mutually exclusive with WAIT and STOPSNAP.

STOPSNAP

STOPSNAP specifies that COPYCHECK is to stop snap all copy relationships initiated in a corresponding COPY step. If COPY must be rerun, STOPSNAP will eliminate the otherwise required wait time for one set of volume copies to complete before copies involving the same volumes can be initiated again.

- Default: None
- Required: Yes
- Restrictions: Mutually exclusive with WAIT and WITHDRAW.

COPYCHECK step JCL example

This topic contains an example of COPYCHECK step JCL. Sample JCL can be found in the installation library SCKZJCL in member CKZCPYCK.

For completeness and to illustrate where JCL parameters and DB2 Cloning Tool control statements must match, the following JCL includes sample DB2 Cloning Tool control statements.

The COPYCHECK step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

1 //?????? JOB , 'CKZ COPYCHECK', CLASS=A, MSGCLASS=X
2 //S1      EXEC PGM=CKZ00010, REGION=8M
3 //STEPLIB DD DSN=HLQ?.SCKZLOAD, DISP=SHR
4 //CKZINI  DD DSN=HLQ?.SCKZPARM(CKZINI), DISP=SHR
5 //CKZPRINT DD SYSOUT=*
6 //SYSUDUMP DD SYSOUT=*
7 //JOURNAL DD DSN=CKZ.JRNL, DISP=OLD
8 //CKZIN   DD *
9           COPYCHECK
10          WAIT(20)
11          JOURNAL-DDN(JOURNAL)
12 //*

```

1. Execution of DB2 Cloning Tool main program.
2. DB2 Cloning Tool SCKZLOAD library must be authorized.
3. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the DB2 Cloning Tool programs.
4. DD for CKZPRINT output.
5. Journal data set used by DB2 Cloning Tool commands. This VSAM data set is used to pass information between DB2 Cloning Tool steps, and to log information derived from the step executions. The journal file is allocated and cataloged in the COPY step and used as input and output by the RENAME step. The specified data set name must match the data set allocated in the COPY step. If multiple DB2 Cloning Tool setups are used for different groups of volumes, take care in specifying the journal data sets for each setup. In the sample JCL, the control statement JOURNAL-DDN(JOURNAL) specifies that a DD with the name JOURNAL is being used rather than a data set name.

DB2ALTERBSDS

This command is optional and is used to alter the contents of a target DB2 BSDS beyond what DB2UPDATE does if any of the following functionality is desired.

- Add a cold start conditional restart record in the BSDS – The keyword COLD-START will create a cold start conditional restart record in the BSDS. The RBA for the cold start will be the next highest 4k value greater than the current high written RBA.
- Delete the DB2 archive log names in the BSDS that are not on the source volumes – The keyword REMOVE-ARCHIVE-LOGS(NOTRENAMED) can remove the DB2 archive log names in the BSDS that are not on the source volumes instead of leaving them in there not renamed.
- Delete non-cloned DB2 active log names in the BSDS – The keyword REMOVE-ACTIVE-LOGS(RETAIN(n)) will remove all of the active logs except n, where n is the number that is selected. The logs will be deleted in start RBA order with the log with the lowest start RBA being deleted first.
- Add a system-level backup (SLB) start conditional restart record in the BSDS – The keyword SLB-START will create an SLB start conditional restart record in the BSDS. The ENDLRSN value that is used in the conditional restart record for the SLB start comes from the System Backup record in the BSDS extracted by the prior DB2UPDATE. This option addresses situations when you are cloning from a DB2 BACKUP SYSTEM and either an active log is defined with more than one stripe or it is a data sharing group. For these cases, the active logs must be truncated at the point when the DB2 BACKUP SYSTEM FlashCopy of the database volumes completed. For data sharing, DB2ALTERBSDS SLB-START should be run for each member of the target data sharing group before the first start of the target member.

The active log will be checked to verify that the SLB LRSN exists for this DB2 subsystem. The conditional restart record will not be created if the active log does not contain the SLB LRSN. This prevents the DB2 subsystem from getting the DSNJ098E CRCL ENDLRSN OF rrr IS HIGHER THAN ANY KNOWN LRSN error message during startup when the active log does not contain the SLB LRSN. This might occur if a member of a data sharing group is infrequently used; it is possible that its active log does not contain log records that are written after the BACKUP SYSTEM data complete LRSN.

This command can be run only after the DB2UPDATE command has been run against the BSDS.

Note: DB2 Cloning Tool commands DB2ALTERBSDS, DB2LGRNXCLEAN, DB2RBLDBSDS, DB2SQL, DB2UPDATE, DB2UTILXCLEAN, and DB2XCFCLEAN must use the same value for DB2-NAME to correctly process for this DB2 subsystem or data-sharing group.

DB2ALTERBSDS command syntax

DB2ALTERBSDS

Required keywords:

{ JOURNAL-DSN(*data set name*) | JOURNAL-DDN(*ddname*) }

Optional keywords:

COLD-START
DB2-MEMBER(*member-name*)
DB2-NAME (*name*)

REMOVE-ACTIVE-LOGS(RETAIN(*nnn*))
REMOVE-ARCHIVE-LOGS(NOTRENAMED)
SIMULATE
SLB-START

DB2ALTERBSDS command and keyword definitions

Required keywords are described first, followed by optional keywords.

DB2ALTERBSDS

Optional command to alter the contents of a target BSDS beyond what DB2UPDATE does.

- Required: No
- Restrictions: This command can be run only after the DB2UPDATE command has been run against the BSDS.

JOURNAL-DSN (*data set name*)

or JOURNAL-DDN (*ddname*)

This parameter supplies either the data set name of the DB2 Cloning Tool journal file or the DD name of the DD statement in the JCL that points to the DB2 Cloning Tool journal file.

DB2ALTERBSDS, whether used to wait for copy completions or to withdraw copy relationships, relies on the volume pairs that are carried in the journal data set from a previously run COPY command.

If multiple DB2 Cloning Tool setups are used for different volume groups, DO NOT use the same journal data set. Each DB2 Cloning Tool "application" needs a different journal data set.

The journal is used to pass information between DB2 Cloning Tool steps (for example, from the COPY step to the RENAME step). Therefore, as noted in the JCL comments, it must be cataloged in the COPY step when first created, and referenced as OLD in subsequent steps.

Do not delete the data set in the last step, in case restarts or reruns need journal information. It is preferable to delete and replace the data set only at the fresh start of a COPY.

- Default: None
- Required: Yes
- Restrictions: None

COLD-START

Specifies that a cold start conditional restart record is to be added to the BSDSs.

Using the REPLY-TO-RESTART-WTOR(Y) keyword with the DB2START command causes DB2START to automatically reply Y to the DB2 WTOR that is a result of the conditional restart record that is created by COLD-START.

- Default: None
- Required: No
- Restrictions: Mutually exclusive with SLB-START.

DB2-MEMBER(*member-name*)

This parameter supplies the member name of the BSDS pair to be altered.

- Default: None
- Required: No
- Restrictions: This parameter is only used for DB2 data sharing.

DB2-NAME(*name*)

Specifies a name that is the same name as the one used for the prior DB2UPDATE for this DB2 subsystem or data-sharing group member.

Note: DB2 Cloning Tool commands DB2ALTERBSDS, DB2LGRNXCLEAN, DB2RBLDBSDS, DB2SQL, DB2UPDATE, DB2UTILXCLEAN, and DB2XCFCLEAN must use the same value for DB2-NAME to correctly process for this DB2 subsystem or data-sharing group.

DB2-NAME is intended to be used when more than one DB2 subsystem or data-sharing group is being cloned from the same DB2 Cloning Tool COPY and RENAME.

Name can be 1 - 4 alpha-numeric-national characters.

- Default: None
- Required: No
- Restrictions: None

REMOVE-ACTIVE-LOGS(RETAIN(*nnn*))

Specifies that active logs are to be removed from the BSDSs.

RETAIN(*nnn*) specifies that all but *nnn* active logs will be removed from the BSDS.

The active logs are removed in start RBA sequence where the active log with the lowest RBA will be deleted first. Care should be taken so there are enough active logs in the BSDS so that DB2 starts up.

- Default: None
- Required: No
- Restrictions: None

REMOVE-ARCHIVE-LOGS(NOTRENAMED)

Specifies that archive logs are to be removed from the BSDSs.

NOTRENAMED specifies that archive logs that were not renamed will be removed.

- Default: None
- Required: No
- Restrictions: None

SIMULATE

Specifies that the actions alter the BSDS are printed as control cards to the DB2 DSNJU003 utility, but no modifications are made to the target BSDSs.

- Default: None
- Required: No
- Restrictions: If SIMULATE is not specified, the previous DB2UPDATE must not have been a SIMULATE.
- Short form: SIM

SLB-START

Specifies that a system-level backup (SLB) start conditional restart record is to be added to the BSDS. The SLB start conditional restart record that is created will have an ENDLRSN value that comes from the system backup record in the BSDS that was extracted by the prior DB2UPDATE.

SLB-START is needed when you are cloning from a DB2 BACKUP SYSTEM and either an active log is defined with more than one stripe or it is a data sharing group. For these cases, the active logs must be truncated

at the point when the DB2 BACKUP SYSTEM FlashCopy of the database volumes completed. For data sharing, DB2ALTERBSDS SLB-START should be run for each member of the target data sharing group before the first start of the target member.

Using the REPLY-TO-RESTART-WTOR(Y) keyword with the DB2START command causes DB2START to automatically reply Y to the DB2 WTOR that is a result of the conditional restart record that is created by SLB-START.

- Default: None
- Required: No
- Restrictions: Mutually exclusive with COLD-START.

DB2ALTERBSDS step JCL example

| Three DB2ALTERBSDS step JCL examples are included: an example to create cold
| start conditional restart record, an example to remove not renamed archive logs,
| and an example to remove some active logs. Sample JCL can be found in the
| installation library SCKZJCL in member CKZDALBS.

DB2ALTERBSDS Step JCL – example to create cold start conditional restart record

For completeness and to illustrate where JCL parameters and DB2 Cloning Tool control statements must match, the following JCL includes sample DB2 Cloning Tool control statements.

The DB2ALTERBSDS step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

    /??????? JOB , 'DB2ALTERBSDS', CLASS=A, MSGCLASS=X
1 //S1      EXEC PGM=CKZ00010, REGION=8M
2 //STEPLIB DD DSN=HLQ?.SCKZLOAD, DISP=SHR
3 //       DD DSN=DSNxxx.SDSNLOAD, DISP=SHR
4 //CKZINI  DD DSN=HLQ?.SCKZPARM(CKZINI), DISP=SHR
5 //CKZPRINT DD SYSOUT=*
    //SORTMSG DD SYSOUT=*
    //SYSUDUMP DD SYSOUT=*
6 //JOURNAL DD DSN=CKZ.JRNL, DISP=OLD
    //CKZIN  DD *
    DB2ALTERBSDS
7     COLD-START
8     DB2-MEMBER(DB1T)
6     JOURNAL-DDN(JOURNAL)
    //*
```

1. Execution of DB2 Cloning Tool main program.
2. DB2 Cloning Tool LOAD library must be authorized.
3. DB2 LOAD library must be authorized.
4. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the DB2 Cloning Tool programs.
5. DD for CKZPRINT output.
6. Journal data set used by DB2 Cloning Tool commands. This VSAM data set is used to pass information between DB2 Cloning Tool steps, and to log information derived from the step executions. The journal file is allocated and cataloged in the COPY step and used as input and output by the RENAME

step. The specified data set name must match the data set allocated in the COPY step. If multiple DB2 Cloning Tool setups are used for different groups of volumes, take care in specifying the journal data sets for each setup. In the sample JCL, the control statement JOURNAL-DDN(JOURNAL) specifies that a DD with the name JOURNAL is being used rather than a data set name.

7. COLD-START parameter that specifies a cold start conditional restart records is to be created.
8. DB2-MEMBER parameter specifying DB2 data-sharing group member name, DB1T, is to have its BSDSs updated.

DB2ALTERBSDS Step JCL – example to remove not renamed archive logs

For completeness and to illustrate where JCL parameters and DB2 Cloning Tool control statements must match, the following JCL includes sample DB2 Cloning Tool control statements.

The DB2ALTERBSDS step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

    //??????? JOB , 'DB2ALTERBSDS', CLASS=A, MSGCLASS=X
1 //S1      EXEC PGM=CKZ00010, REGION=8M
2 //STEPLIB DD DSN=HLQ?.SCKZLOAD, DISP=SHR
3 //      DD DSN=DSNxxx.SDSNLOAD, DISP=SHR
4 //CKZINI  DD DSN=HLQ?.SCKZPARM(CKZINI), DISP=SHR
5 //CKZPRINT DD SYSOUT=*
  //SORTMSG DD SYSOUT=*
  //SYSUDUMP DD SYSOUT=*
6 //JOURNAL DD DSN=CKZ.JRNL, DISP=OLD
  //CKZIN   DD *
    DB2ALTERBSDS
7     REMOVE-ARCHIVE-LOGS(NOTRENAMED)
8     DB2-MEMBER(DB1T)
6     JOURNAL-DDN(JOURNAL)
  //*
```

1. Execution of DB2 Cloning Tool main program.
2. DB2 Cloning Tool LOAD library must be authorized.
3. DB2 LOAD library must be authorized.
4. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the DB2 Cloning Tool programs.
5. DD for CKZPRINT output.
6. Journal data set used by DB2 Cloning Tool commands. This VSAM data set is used to pass information between DB2 Cloning Tool steps, and to log information derived from the step executions. The journal file is allocated and cataloged in the COPY step and used as input and output by the RENAME step. The specified data set name must match the data set allocated in the COPY step. If multiple DB2 Cloning Tool setups are used for different groups of volumes, take care in specifying the journal data sets for each setup. In the sample JCL, the control statement JOURNAL-DDN(JOURNAL) specifies that a DD with the name JOURNAL is being used rather than a data set name.
7. REMOVE-ARCHIVE-LOGS parameter that specifies all not renamed archive log records be removed.
8. DB2-MEMBER parameter specifying DB2 data-sharing group member name, DB1T, is to have its BSDSs updated.

DB2ALTERBSDS Step JCL – Example to Remove Some Active Logs

For completeness and to illustrate where JCL parameters and DB2 Cloning Tool control statements must match, the following JCL includes sample DB2 Cloning Tool control statements.

The DB2ALTERBSDS step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

  //??????? JOB , 'DB2ALTERBSDS' , CLASS=A,MSGCLASS=X
1 //S1      EXEC PGM=CKZ00010,REGION=8M
2 //STEPLIB DD DSN=HLQ?.SCKZLOAD,DISP=SHR
3 //        DD DSN=DSNxxx.SDSNLOAD,DISP=SHR
4 //CKZINI  DD DSN=HLQ?.SCKZPARM(CKZINI) ,DISP=SHR
5 //CKZPRINT DD SYSOUT=*
  //SORTMSG DD SYSOUT=*
  //SYSUDUMP DD SYSOUT=*
6 //JOURNAL DD DSN=CKZ.JRNL,DISP=OLD
  //CKZIN   DD *
    DB2ALTERBSDS
7     REMOVE-ACTIVE-LOGS(RETAIN(3))
8     DB2-MEMBER(DB1T)
6     JOURNAL-DDN(JOURNAL)
  //*
```

1. Execution of DB2 Cloning Tool main program.
2. DB2 Cloning Tool LOAD library must be authorized.
3. DB2 LOAD library must be authorized.
4. DD for DB2 Cloning Tool.SCKZPARM, CKZINI member. The CKZINI member is used to provide variables to the DB2 Cloning Tool programs.
5. DD for CKZPRINT output.
6. Journal data set used by DB2 Cloning Tool commands. This VSAM data set is used to pass information between DB2 Cloning Tool steps, and to log information derived from the step executions. The journal file is allocated and cataloged in the COPY step and used as input and output by the RENAME step. The specified data set name must match the data set allocated in the COPY step. If multiple DB2 Cloning Tool setups are used for different groups of volumes, take care in specifying the journal data sets for each setup. In the sample JCL, the control statement JOURNAL-DDN(JOURNAL) specifies that a DD with the name JOURNAL is being used rather than a data set name.
7. REMOVE-ACTIVE-LOGS parameter that specifies to remove all but the most recent 3 active logs.
8. DB2-MEMBER parameter specifying DB2 data-sharing group member name, DB1T, is to have its BSDSs updated.

DB2FIX

This command is not required. This command is only used for 'online' cloning of a DB2 subsystem. An online cloning is when the source DB2 subsystem is active (online) at the time the source volumes were copied. DB2FIX will fix target DB2 page spaces that have LPL or GRECP status by issuing a DB2 START DATABASE command against them. If the DB2 system is data sharing, only one DB2 member should be running when DB2FIX is run.

The DB2FIX command should be run twice. The first run will fix any DB2 CATALOG (DSNDB06) or DB2 Directory (DSNDB01) page spaces by using DATABASES(DB2) and the second run will fix all other page spaces by using DATABASES(APPLICATION). The second run with DATABASES(APPLICATION) must only happen after the DB2 catalog has been updated with the DB2SQL command.

When using the keyword DATABASES(DB2), if page space DBD01 (SYSDBDXA for DB2 10 NFM) in database DSNDB01 was started due to having LPL or GRECP status, DB2FIX will: 1) complete with the return code specified in the DSNDB01-DBD01-STARTED, MEMBERS-AND-DBD01, or WAIT-AND-DBD01 keywords, and 2) display the following WTO message:

```
CKZ23526W DSNDB01.DBD01 IS IN RESTRICTED STATUS; DB2UPDATE NEEDS TO BE RUN AGAIN
```

For DB2 10 NFM the message is:

```
CKZ23526W DSNDB01.SYSDBDXA IS IN RESTRICTED STATUS; DB2UPDATE NEEDS TO BE RUN AGAIN
```

If this error happens, the changes made to DBD01 (SYSDBDXA for DB2 10 NFM) by DB2UPDATE may have been regressed and need to be redone. The target DB2 subsystem must be stopped, DB2UPDATE run again using the DBD01ONLY keyword, and the DB2 subsystem started again in maintenance mode using the special zparms, prior to running DB2SQL and DB2FIX using DATABASES(APPLICATION).

These steps can be automated by using either the DB2FIX return code setting, or have an automated operations package monitor the DB2FIX WTO message produced when DB2FIX starts restricted page spaces.

If not all data is being cloned with the DB2 subsystem, it is possible that some of the table and index spaces that are not cloned will have LPL or GRECP status. DB2FIX will receive a timeout on the start commands issued for these page spaces. The EXCLUDE-MASKS keyword can be used to cause DB2FIX to not issue start commands for the page spaces that are not cloned.

For data sharing, there is a possibility that other members will hold locks on table and index spaces that have LPL or GRECP status. Normally the locks are released when the other members are initially started in special and light mode. If some locks are not released by a member after the initial start, it will be necessary to start that member in special and not light mode so it remains running. Then running DB2FIX with MEMBERS-NEED-STARTING(ACTION(CONTINUE)) will usually resolve the LPL or GRECP status. If DB2FIX is still unable to resolve the LPL or GRECP status, then running DB2FIX with MEMBERS-NEED-STARTING(ACTION(CONTINUE)) on each active member will usually resolve the LPL or GRECP status.

The userid that runs the DB2FIX command must be defined in the special zparms as either SYSADM or SYSADM2.

For more information on this command, see the following topics:

- “Status of transactions in flight” on page 104

- “DB2 online cloning with removal of data sharing members procedure” on page 111
- “DB2 online cloning with target becoming non-data sharing procedure” on page 117

DB2FIX command syntax

DB2FIX

Required keywords:

DATABASES(DB2 | APPLICATION)
DB2-SSID(*db2 ssid*)

Optional keywords:

DSNDB01-DBD01-STARTED(RC(*nn* | 16))
EXCLUDE-MASKS(*masks*)
MAX-CONCURRENT-CMDS(*nn* | 1)
MEMBERS-AND-DBD01(RC(*nn* | 16))
MEMBERS-NEED-STARTING([RC(*nn* | 8)] [ACTION(QUIT | CONTINUE)])
SIMULATE
START-SCOPE(DATABASE | PAGESPACE)
WAIT(*nnn* | 5 [,RC(*rr* | 8)] [ACTION(QUIT | CONTINUE)])
WAIT-AND-DBD01(RC(*nn* | 16))

EXCLUDE-MASKS Considerations

The exclude mask is used to select the table and index spaces that will not be started by DB2FIX. The format of an exclude mask entry is dbname.spname, where dbname is the database name and spname is the space name. The allowable filter characters are shown in the following table:

Table 54. Filter characters allowed for filter exclude mask syntax

Character	Description
*	A single asterisk represents 0 to nn characters of any value.
%	A percent sign represents one non-blank character.
!	An exclamation point represents one national character. @ # \$
<	A lesser-than sign represents one non-numeric character, national symbols included.
>	A greater-than sign represents one numeric character.

For example, exclude mask = APPL1DB.* would match all spaces in database APPL1DB.

For information about filters and ACS masks, refer to the topic “Filtering pattern masks” on page 23.

DB2FIX command and keyword definitions

Required keywords are described first, followed by optional keywords.

DB2FIX

Optional command to correct DB2 page spaces that are in restricted status.

- Required: No
- Restrictions: This command is only used for ‘online’ cloning of a DB2 subsystem with DB2 SET LOG SUSPEND.

DATABASES(DB2 | APPLICATION)

This parameter specifies which database page spaces will be fixed.

DB2 specifies that page spaces in the DB2 catalog (DSNDB06) and DB2 Directory (DSNDB01) are to be fixed.

APPLICATION specifies that page spaces in all other databases (not DSNDB01 and DSNDB06) are to be fixed. The running of DB2FIX with DATABASES(APPLICATION) must only be done after the running of the DB2SQL command.

- Default: None
- Required: Yes
- Restrictions: None

DB2-SSID(*db2 ssid*)

This parameter supplies the DB2 SSID of the DB2 subsystem to connect to. A group name should not be used for this parameter.

- Default: None
- Required: Yes
- Restrictions: None

DSNDB01-DBD01-STARTED(RC(*nm* | 16))

This parameter supplies the return code that will be used if table space DBD01 (SYSDBDXA for DB2 10 NFM) in database DSNDB01 was started due to the table space having LPL or GRECP status.

- Default: RC(16)
- Required: No
- Restrictions: None.

EXCLUDE-MASKS(*masks*)

This parameter supplies a list of masks of the form dbname.spname, where dbname is the database name and spname is the space name. If a database and page space name matches an entry in the list, no start command will be issued for it.

If START-SCOPE(DATABASE) is also specified, page space names that match an entry in the EXCLUDE-MASKS list will still be started if there are page spaces in the same database that are not in the list.

- Default: None
- Required: No
- Restrictions: Applies only when DATABASES(APPLICATION) is specified.

MAX-CONCURRENT-CMDS(*nm* | 1)

Specifies the maximum number of start commands that will be processed concurrently. Increasing this number can decrease the elapsed time of DB2FIX and will increase the resources being used by DB2. Experiment with this parameter to determine the optimum setting.

- Default: 1
- Required: No
- Restrictions: Applies only when DATABASES(APPLICATION) is specified.

MEMBERS-AND-DBD01(RC(*nm* | 16))

This parameter supplies the return code that will be used if table space DBD01 (SYSDBDXA for DB2 10 NFM) in database DSNDB01 was started

(due to the table space having LPL or GRECP status), and if other members in this data-sharing group need to be started because they hold locks.

- Default: RC(16)
- Required: No
- Restrictions: None

MEMBERS-NEED-STARTING([RC(*mm* | 8)] [, ACTION(QUIT | CONTINUE)])

This parameter supplies the return code that will be used and the action to be taken if other members in this data-sharing group need to be started because they hold locks. The ACTION parameter specifies whether to QUIT processing or to CONTINUE processing by attempting to start restricted objects when locks are held by other members.

ACTION(CONTINUE) is used to cause DB2FIX to issue the START DATABASE commands for table and index spaces in LPL or GRECP status even though other members in the data sharing group have locks outstanding. If a timeout occurs when using ACTION(CONTINUE), it may be necessary to run DB2FIX using ACTION(CONTINUE) on the other members as well. This option addresses the situation for data sharing where other members are reported to hold locks even after they have been started in light mode.

- Default: RC(8), ACTION(QUIT)
- Required: No
- Restrictions: None

SIMULATE

Specifies that DB2FIX will show what it will do, but will not issue any DB2 START DB commands.

- Default: None
- Required: No
- Restrictions: None
- Short form: SIM

START-SCOPE(DATABASE | PAGESPACE)

Specifies the scope of the DB2 start database commands that will be used.

DATABASE specifies that a single DB2 start database command will be done for all page spaces in a database that have LPL or GRECP status (e.g. STA DB(dbname) SPACENAM(*)).

PAGESPACE specifies that a single DB2 start database command will be done for each page space in the database that has LPL or GRECP status (e.g. STA DB(dbname) SPACENAM(spacename)).

- Default: PAGESPACE
- Required: No
- Restrictions: Applies only when DATABASES(APPLICATION) is specified.

WAIT(*nnn* | 5 [,RC(*rr* | 8)] [, ACTION(QUIT | CONTINUE)])

nnn specifies the maximum time in minutes that DB2FIX will wait for a single DB2 start database command to fix the page spaces that were in LPL or GRECP status. If the specified time limit expires before the DB2 start database command has fixed the page spaces in LPL or GRECP status,

DB2FIX will terminate with return code rr. When a timeout occurs, the ACTION parameter specifies whether to QUIT processing or CONTINUE processing.

- Default: 5,RC(8), ACTION(QUIT)
- Required: No
- Restrictions: ACTION(CONTINUE) applies only when DATABASES(APPLICATION) is specified.

WAIT-AND-DBD01(RC(*nn* | 16))

This parameter supplies the return code that will be used if table space DBD01 (SYSDBDXA for DB2 10 NFM) in database DSNDB01 was started due to its being found to have LPL or GRECP status and if the WAIT time limit was exceeded for a DB2 start database command to fix page spaces in LPL or GRECP status.

- Default: RC(16)
- Required: No
- Restrictions: None.

DB2FIX step JCL example

This topic contains an example of DB2FIX step JCL. Sample JCL can be found in the installation library SCKZJCL in member CKZDFIX.

For completeness and to illustrate where JCL parameters and DB2 Cloning Tool control statements must match, the following JCL includes sample DB2 Cloning Tool control statements.

The DB2FIX step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

      //??????? JOB , 'DB2FIX', CLASS=A, MSGCLASS=X
1 //S1      EXEC PGM=CKZ00010, REGION=8M
2 //STEPLIB DD DSN=HLQ?.SCKZLOAD, DISP=SHR
3 //        DD DSN=DSNxxx.SDSNLOAD, DISP=SHR
4 //CKZINI  DD DSN=HLQ?.SCKZPARAM(CKZINI), DISP=SHR
5 //CKZPRINT DD SYSOUT=*
  //SYSUDUMP DD SYSOUT=*
  //CKZIN   DD *
      DB2FIX          -
6      DB2-SSID(DB1T) -
7      DATABASES(DB2)
  //*
```

1. Execution of DB2 Cloning Tool main program.
2. DB2 Cloning Tool LOAD library must be authorized.
3. DB2 LOAD library must be authorized.
4. DD for CKZINI, SCKZPARAM member. The CKZINI member of the HLQ?.SCKZPARAM library provides variables to the DB2 Cloning Tool programs.
5. DD for CKZPRINT output.
6. DB2-SSID parameter specifying DB2 ssid, DB1T, that will be connected to.
7. DATABASES parameter specifying the databases, DB2, that will be checked for LPL and GRECP status and started as necessary.

DB2GETBACKINFO

This command is not required. DB2GETBACKINFO will issue HSM LIST COPYPOOL commands to determine the available backups from DB2 BACKUP SYSTEM and read the results to determine the source and backup volume pairs to be used. The source volume to backup volume pairs and optionally source ICF catalog information will be written to the backinfo data set.

This command is used as part of the process to clone from a DB2 BACKUP SYSTEM backup or backup dump tapes. Detailed descriptions of these cloning processes can be found in “Cloning scenarios” on page 535.

Using the LIST option will cause information about all backups or a specific backup to be displayed for a specific DB2 system.

Information about backups is obtained from HSM by issuing commands of the form:

```
LIST COPYPOOL(DSN$location$type) ODS(workdsn)
```

Where:

- type is DB for the database copy pool or LG for the log copy pool.
- The location used comes from the LOCATION keyword, or is read from a BSDS if the BSDS-DDN keyword is used.
- The workdsn data set name comes from the WORK-DDN or WORK-DSN keyword. This data set cannot be a temporary data set.

HSM writes the output of the list command to the workdsn data set. DB2GETBACKINFO then reads and parses the data set to identify the backups and their information.

The use of the HSM LIST COPYPOOL command can be protected by RACF with a profile in the FACILITY class of the form STGADMIN.ARC.LC.copypoolname. The userid that runs the DB2GETBACKINFO command must have at least READ access to such a profile if it exists.

The work data set identified by the WORK-DDN or WORK-DSN keyword is updated by both HSM and the DB2BACKINFO command. HSM and the userid that runs the DB2GETBACKINFO command must have at least UPDATE authority to this data set.

DB2GETBACKINFO command syntax

DB2GETBACKINFO

Required keywords:

```
{ BSDS-DDN( ddname ) | LOCATION( location ) }  
{ WORK-DDN( ddname ) | WORK-DSN( dsname ) }
```

Optional keywords:

```
BACKINFO-DDN( ddname )  
{ TOKEN( X'token' ) | LAST }  
LIST  
{ LOGSONLY | DATABASESONLY }  
USE-DUMPTAPES  
DUMP-CLASS( classname )  
USERCATALOGS( sourcecat1, ..., sourcecatn )
```

DB2GETBACKINFO command and keyword definitions

Required keywords are described first, followed by optional keywords.

DB2GETBACKINFO

Optional command to allow for easier cloning from a DB2 BACKUP SYSTEM FULL backup.

- Required: No.
- Restrictions: None.

BSDS-DDN(*ddname*)

This parameter specifies the DD name that points to a BSDS from which the location will be obtained. The location will be used in the HSM copy pool names.

- Default: None.
- Required: Yes.
- Restrictions: Mutually exclusive with LOCATION.

LOCATION(*location*)

This parameter specifies the location that will be used in the HSM copy pool names.

- Default: None.
- Required: Yes.
- Restrictions: Mutually exclusive with BSDS-DDN.

WORK-DDN(*ddname*)

This parameter specifies the DD name that points to the HSM list data set that is updated by HSM LIST commands and read by the DB2GETBACKINFO command. The data set allocated to this DD cannot be a temporary data set.

- Default: None.
- Required: No.
- Restrictions: Mutually exclusive with WORK-DSN.

WORK-DSN(*dsname*)

This parameter specifies the data set name of the HSM list data set that is updated by HSM LIST commands and read by the DB2GETBACKINFO command.

- Default: None.
- Required: Yes.
- Restrictions: Mutually exclusive with WORK-DDN.

BACKINFO-DDN(*ddname*)

This parameter specifies the DD name which points to a file where the backinfo information will be written.

- Default: None.
- Required: Required if LIST is not specified.
- Restrictions: Not used if LIST is specified.

DATABASESONLY

This parameter specifies that only the database (DB) volumes of the selected backup should be used. DATABASESONLY is intended to be used when you want to clone only the database volumes of the selected backup.

- Default: None.
- Required: No.

- Restrictions: Mutually exclusive with LOGSONLY.
- Short form: DBONLY

DUMP-CLASS(*classname*)

This parameter specifies the tape dump class to identify the correct tape volumes. This option addresses the situation when there are multiple dump classes for a DB2 BACKUP SYSTEM.

- Default: None.
- Required: Only required if USE-DUMPTAPES is specified and there is more than one tape dump class used for the backup.
- Restrictions: Only valid if USE-DUMPTAPES is specified.

LAST This parameter specifies that the last (most recent) backup is to be used.

- Default: None.
- Required: Required if LIST is not specified.
- Restrictions: Mutually exclusive with TOKEN.

LIST This parameter specifies that a list be generated. If LAST or TOKEN are also specified only one backup, corresponding to the last or the token value, will be listed.

- Default: None.
- Required: No.
- Restrictions: None.

LOGSONLY

This parameter specifies that only the log (LG) volumes of the selected backup should be used. LOGSONLY is intended to be used when you want to clone only the log volumes of the selected backup.

- Default: None.
- Required: No.
- Restrictions: Mutually exclusive with DATABASESONLY.
- Short form: LGONLY

TOKEN(X'*token*')

This parameter specifies that a specific backup is to be used. The value of token to use can be obtained by using the LIST option of this command, using a HSM LIST COPYPOOL command, or running the DSNJU004 utility.

- Default: None.
- Required: Required if LIST is not specified.
- Restrictions: Mutually exclusive with LAST.

USE-DUMPTAPES

Specify this parameter to retrieve information about system-level backup dump tape volumes instead of the backup volumes. The information from HSM is used to populate the backinfo data set.

When you specify USE-DUMPTAPES, DB2GETBACKINFO excludes the following from consideration:

- Backups with an HSM DUMPSTATE that is not ALLCOMPLETE
- Backups with at least one dump tapes that have an expiration date prior to the current date
- Default: None.
- Required: No.

- Restrictions: None.

USERCATALOGS(*srccat1, ..., srccatn*)

This parameter specifies the source ICF catalogs that are being used by the source DB2 system. It is required that all the specified ICF catalogs reside on source volumes.

- Default: None.
- Required: No.
- Restrictions: None.

DB2GETBACKINFO step JCL examples

This topic contains examples of DB2GETBACKINFO step JCL. Sample JCL can be found in the installation library SCKZJCL in member CKZDGETB.

For completeness and to illustrate where JCL parameters and DB2 Cloning Tool control statements must match, the following JCL includes sample DB2 Cloning Tool control statements.

The DB2GETBACKINFO step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

1 //??????? JOB , 'CKZ DB2GETBACKINFO', CLASS=A, MSGCLASS=X
2 //S0      EXEC PGM=IDCAMS
3 //SYSPRINT DD  SYSOUT=*
4 //SYSIN   DD  *
5 1  DEL CKZ.WRK.BACKINFO
6 2  DEL CKZ.WRK.HSMLIST
7   SET MAXCC=0
8 //S1      EXEC PGM=CKZ00010, REGION=8M
9 //STEPLIB DD  DSN=HLQ?.SCKZLOAD, DISP=SHR
10 //CKZINI  DD  DSN=HLQ?.SCKZPARM(CKZINI), DISP=SHR
11 //CKZPRINT DD  SYSOUT=*
12 //SYSUDUMP DD  SYSOUT=*
13 //BACKINFO DD  DSN=CKZ.WRK.BACKINFO,
14 //          DISP=(,CATLG), UNIT=SYSALLDA,
15 //          SPACE=(CYL,(1,1))
16 //HSMLIST  DD  DSN=CKZ.WRK.HSMLIST,
17 //          DISP=(,CATLG), UNIT=SYSALLDA,
18 //          SPACE=(CYL,(1,1))
19 //CKZIN   DD  *
20   DB2GETBACKINFO
21   BACKINFO-DDN(BACKINFO)
22   WORK-DDN(HSMLIST)
23   LAST
24   LOCATION(DB2PLOC)
25   USERCATALOGS(
26     USERCAT.SRC01
27     USERCAT.SRC02
28   )
29 //*

```

1. Deletion of backinfo data set in anticipation of allocating new for each execution.
2. Deletion of HSM list data set in anticipation of allocating new for each execution.
3. Execution of DB2 Cloning Tool main program.
4. DB2 Cloning Tool LOAD library must be authorized.

5. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the DB2 Cloning Tool programs.
6. DD for CKZPRINT output.
7. The backinfo data set that is created by DB2GETBACKINFO and will be used by a subsequent DB2 Cloning Tool BACKINFO-REFORMAT command. In the sample JCL, the control statement BACKINFO-DDN(BACKINFO) specifies that a DD statement with the name BACKINFO is used.
8. The HSM list data set that is updated by HSM LIST commands and read by the DB2GETBACKINFO command. In the sample JCL, the control statement WORK-DDN(HSMLIST) specifies that a DD statement with the name HSMLIST is used to identify the work data set. This data set cannot be a temporary data set.
9. Specifies that the last DB2 BACKUP SYSTEM backup be used.
10. The location that identifies the DB2 whose BACKUP SYSTEM backup is to be used.
11. The source ICF catalogs that are being used by the source DB2 system. These ICF catalogs must reside on source volumes.

Example of using DB2GETBACKINFO to get BACKUP SYSTEM dump tape information

For completeness and to illustrate where JCL parameters and DB2 Cloning Tool control statements must match, the following JCL includes sample DB2 Cloning Tool control statements. Sample JCL can be found in the installation library SCKZJCL in member CKZDGETB.

The DB2GETBACKINFO step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

1  //??????? JOB , 'CKZ DB2GETBACKINFO', CLASS=A, MSGCLASS=X
2  //S0      EXEC PGM=IDCAMS
3  //SYSPRINT DD  SYSOUT=*
4  //SYSIN   DD  *
5  1  DEL CKZ.WRK.BACKINFO
6  2  DEL CKZ.WRK.HSMLIST
7  SET MAXCC=0
8  //S2      EXEC PGM=CKZ00010, REGION=8M, PARM=' '
9  //*
10 //STEPLIB DD  DISP=SHR, DSN=HLQ?.LOADLIB
11 //CKZINI  DD  DISP=SHR, DSN= HLQ?.SCKZPARM(CKZINI)
12 //CKZPRINT DD  SYSOUT=*
13 //BACKINFO DD  DISP=(,CATLG), DSN=HLQ?.BACKINFO,
14 //          UNIT=SYSALLDA, SPACE=(TRK, (3,3))
15 //HSMLIST  DD  DISP=(,CATLG), DSN=HLQ?.HSMLIST,
16 //          UNIT=SYSALLDA, SPACE=(TRK, (3,3))
17 //CKZIN   DD  *
18 DB2GETBACKINFO -
19 USE-DUMPTAPES -
20 DUMP-CLASS(DUMPTP) -
21 BACKINFO-DDN(BACKINFO) -
22 WORK-DDN(HSMLIST) -
23 LAST -
24 LOCATION(DB2-LOCATION) -
25 USERCATALOGS( -
26     USERCAT.SRC1 -
27     USERCAT.SRC2 -
28 )
29 //*

```

1. Deletion of backinfo data set in anticipation of allocating new for each execution.
2. Deletion of HSM list data set in anticipation of allocating new for each execution.
3. Execution of DB2 Cloning Tool main program.
4. DB2 Cloning Tool LOAD library must be authorized.
5. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the DB2 Cloning Tool programs.
6. DD for CKZPRINT output.
7. The backinfo data set that is created by DB2GETBACKINFO and will be used by a subsequent DB2 Cloning Tool RESTORE-FROM-DUMPTAPES command. In the sample JCL, the control statement BACKINFO-DDN(BACKINFO) specifies that a DD statement with the name BACKINFO is used. The backinfo data set is input to the command and must be generated by a DB2GETBACKINFO command from the source DB2 system or LPAR. If not on shared DASD with the target system, the backinfo data set must be transferred via FTP or other method to the target system.
8. The HSM list data set that is updated by HSM LIST commands and read by the DB2GETBACKINFO command. In the sample JCL, the control statement WORK-DDN(HSMLIST) specifies that a DD statement with the name HSMLIST is used to identify the work data set. This data set cannot be a temporary data set.
9. Directs DB2GETBACKINFO to look only for dump tapes in the HSM LIST COPYPOOL output. This keyword is required when generating a backinfo data set for a RESTORE-FROM-DUMPTAPES command.
10. Directs DB2GETBACKINFO to search only for the specified DUMP-CLASS in the HSM LIST COPYPOOL output. Replace *DUMPTP* with the actual DUMP-CLASS. This keyword is optional; if not specified and multiple tape dump classes are found, the first one found in the HSM LIST COPYPOOL output is used.
11. Specifies that the last DB2 BACKUP SYSTEM backup be used.
12. The location that identifies the DB2 whose BACKUP SYSTEM backup is to be used.
13. The source ICF catalogs that are being used by the source DB2 system. These ICF catalogs must reside on source volumes.

DB2LGRNXCLEAN

This command is optional. It is used only for the cloning of a DB2 data-sharing group when you want to physically remove members from the target DB2 subsystem.

DB2LGRNXCLEAN removes all entries from SYSLGRNX. For more information about this command, see the following topics:

- “DB2 offline cloning with removal of data sharing members procedure” on page 98
- “DB2 online cloning with removal of data sharing members procedure” on page 111

Note: DB2 Cloning Tool commands DB2ALTERBSDS, DB2LGRNXCLEAN, DB2RBLDBSDS, DB2SQL, DB2UPDATE, DB2UTILXCLEAN, and DB2XCFCLEAN must use the same value for **DB2-NAME** to correctly process for this DB2 subsystem or data-sharing group.

DB2LGRNXCLEAN command syntax

DB2LGRNXCLEAN

Required keywords:

{ JOURNAL-DSN(*data set name*) | JOURNAL-DDN(*ddname*) }

Optional keywords:

DB2-NAME (*name*)
SIMULATE

DB2LGRNXCLEAN command and keyword definitions

Required keywords are described first, followed by optional keywords.

DB2LGRNXCLEAN

Optional command to clean out DB2 SYSLGRNX as part of removing other members.

- Required: No
- Restrictions: Used only for the cloning of a DB2 data-sharing group when you want to physically remove members from the target DB2 subsystem.

JOURNAL-DSN (*data set name*)

or JOURNAL-DDN (*ddname*)

This parameter supplies either the data set name of the DB2 Cloning Tool journal file or the DD name of the DD statement in the JCL that points to the DB2 Cloning Tool journal file.

DB2ALTERBSDS, whether used to wait for copy completions or to withdraw copy relationships, relies on the volume pairs that are carried in the journal data set from a previously run COPY command.

If multiple DB2 Cloning Tool setups are used for different volume groups, DO NOT use the same journal data set. Each DB2 Cloning Tool "application" needs a different journal data set.

The journal is used to pass information between DB2 Cloning Tool steps (for example, from the COPY step to the RENAME step). Therefore, as noted in the JCL comments, it must be cataloged in the COPY step when first created, and referenced as OLD in subsequent steps.

Do not delete the data set in the last step, in case restarts or reruns need journal information. It is preferable to delete and replace the data set only at the fresh start of a COPY.

- Default: None
- Required: Yes
- Restrictions: None

DB2-NAME(*name*)

Specifies a name that is the same name as the one used for the prior DB2UPDATE for this DB2 subsystem or data-sharing group.

Note: DB2 Cloning Tool commands DB2ALTERBSDS, DB2LGRNXCLEAN, DB2RBLDBSDS, DB2SQL, DB2UPDATE, DB2UTILXCLEAN, and DB2XCFCLEAN must use the same value for DB2-NAME to correctly process for this DB2 subsystem or data-sharing group.

DB2-NAME is intended to be used when more than one DB2 subsystem or data-sharing group is being cloned from the same DB2 Cloning Tool COPY and RENAME.

Name can be 1 - 4 alpha-numeric-national characters.

- Default: None
- Required: No
- Restrictions: None

SIMULATE

Specifies that what will be done to clean out DB2 SYSLGRNX is shown, but no modifications are made to SYSLGRNX.

- Default: None
- Required: No
- Restrictions: If SIMULATE is not specified, the previous DB2UPDATE must not have been a SIMULATE.
- Short form: SIM

DB2LGRNXCLEAN step JCL example

This topic contains an example of DB2LGRNXCLEAN step JCL. Sample JCL can be found in the installation library SCKZJCL in member CKZDLGCL.

For completeness and to illustrate where JCL parameters and DB2 Cloning Tool control statements must match, the following JCL includes sample DB2 Cloning Tool control statements.

The DB2LGRNXCLEAN step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

      /??????? JOB , 'DB2LGRNXCLEAN', CLASS=A, MSGCLASS=X
1 //S1      EXEC PGM=CKZ00010, REGION=8M
2 //STEPLIB DD DSN=HLQ?.SCKZLOAD, DISP=SHR
      //      DD DSN=DSNxxx.SDSNLOAD, DISP=SHR
3 //CKZINI  DD DSN=HLQ?.SCKZPARM(CKZINI), DISP=SHR
4 //CKZPRINT DD SYSOUT=*
      //SYSUDUMP DD SYSOUT=*
5 //JOURNAL DD DSN=CKZ.JRNL, DISP=OLD
      //CKZIN  DD *
          DB2LGRNXCLEAN
          JOURNAL-DDN(JOURNAL)
      //*

```

1. Execution of DB2 Cloning Tool main program.
2. DB2 Cloning Tool LOAD library must be authorized.
3. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the DB2 Cloning Tool programs.
4. DD for CKZPRINT output.
5. Journal data set used by DB2 Cloning Tool commands. This VSAM data set is used to pass information between DB2 Cloning Tool steps, and to log information derived from the step executions. The journal file is allocated and

cataloged in the COPY step and used as input and output by the RENAME step. The specified data set name must match the data set allocated in the COPY step. If multiple DB2 Cloning Tool setups are used for different groups of volumes, take care in specifying the journal data sets for each setup. In the sample JCL, the control statement JOURNAL-DDN(JOURNAL) specifies that a DD with the name JOURNAL is being used rather than a data set name.

DB2RBLDBSDS

This command is optional. This command is used only for the cloning of a DB2 data-sharing group when you want to physically remove members from the target BSDS or to make the target DB2 non-data sharing.

This command must be run for every target member that is to be used. For more information about this command, see the following topics:

- “DB2 offline cloning with removal of data sharing members procedure” on page 98
- “DB2 offline cloning with target becoming non-data sharing procedure” on page 101
- “DB2 online cloning with removal of data sharing members procedure” on page 111
- “DB2 online cloning with target becoming non-data sharing procedure” on page 117

Note: DB2 Cloning Tool commands DB2ALTERBSDS, DB2LGRNXCLEAN, DB2RBLDBSDS, DB2SQL, DB2UPDATE, DB2UTILXCLEAN, and DB2XCFCLEAN must use the same value for DB2-NAME to correctly process for this DB2 subsystem or data-sharing group.

DB2RBLDBSDS command syntax

DB2RBLDBSDS

Required keywords:

```
DB2-MEMBER( member name )
{ JOURNAL-DSN( data set name ) | JOURNAL-DDN( ddname ) }
```

Optional keywords:

```
ACTIVELOG-LIMIT( ALL | nnnn )
DATA-SHARING( SOURCE | NO )
DB2-NAME ( name )
DDF( SOURCE | NO )
PRIMARY( SOURCE | YES | NO )
SIMULATE
```

DB2RBLDBSDS command and keyword definitions

Required keywords are described first, followed by optional keywords.

DB2RBLDBSDS

Optional command to rebuild a DB2 BSDS to remove other members or to make it non-data sharing.

- Required: No
- Restrictions: Used only for cloning a DB2 data-sharing group when you want to physically remove members from the target BSDS or to make the target DB2 non-data sharing.

DB2-MEMBER (*member name*)

This parameter supplies the member name of the BSDS pair to be rebuilt.

- Default: None
- Required: No
- Restrictions: None

JOURNAL-DSN (*data set name*)**or JOURNAL-DDN (*ddname*)**

This parameter supplies either the data set name of the DB2 Cloning Tool journal file or the DD name of the DD statement in the JCL that points to the DB2 Cloning Tool journal file.

DB2ALTERBSDS, whether used to wait for copy completions or to withdraw copy relationships, relies on the volume pairs that are carried in the journal data set from a previously run COPY command.

If multiple DB2 Cloning Tool setups are used for different volume groups, DO NOT use the same journal data set. Each DB2 Cloning Tool "application" needs a different journal data set.

The journal is used to pass information between DB2 Cloning Tool steps (for example, from the COPY step to the RENAME step). Therefore, as noted in the JCL comments, it must be cataloged in the COPY step when first created, and referenced as OLD in subsequent steps.

Do not delete the data set in the last step, in case restarts or reruns need journal information. It is preferable to delete and replace the data set only at the fresh start of a COPY.

- Default: None
- Required: Yes
- Restrictions: None

ACTIVELOG-LIMIT (ALL | *nnnn*)

Specifies how many active logs will be retained in the rebuilt target BSDS for this member. ALL specifies that all active logs be retained in the BSDS. *nnnn* specifies the number of active logs to retain in the BSDS. Care should be taken so there are enough active logs in the BSDS so that DB2 starts up.

- Default: ALL
- Required: No
- Restrictions: This parameter is not used if DATA-SHARING(NO) is specified.

DATA-SHARING (SOURCE | NO)

Specifies that the actions to alter the BSDS are printed as control cards to the DB2 DSNJU003 utility, but no modifications are made to the target BSDSs.

- Default: Source
- Required: No
- Restrictions: If DATA-SHARING(NO) is specified, keywords ACTIVELOG-LIMIT, DDF, and PRIMARY are not used.

DB2-NAME(*name*)

Specifies a name that is the same name as the one used for the prior DB2UPDATE for this DB2 subsystem or data-sharing group member.

Note: DB2 Cloning Tool commands DB2ALTERBSDS, DB2LGRNXCLEAN, DB2RBLDBSDS, DB2SQL, DB2UPDATE, DB2UTILXCLEAN, and

DB2XCFCLEAN must use the same value for DB2-NAME to correctly process for this DB2 subsystem or data-sharing group.

DB2-NAME is intended to be used when more than one DB2 subsystem or data-sharing group is being cloned from the same DB2 Cloning Tool COPY and RENAME.

Name can be 1 - 4 alpha-numeric-national characters.

- Default: None
- Required: No
- Restrictions: None

DDF (SOURCE | NO)

Specifies whether the DDF information in the rebuilt BSDS will be retained for this member. SOURCE specifies that the DDF information is retained. NO specifies that the DDF information is not retained.

- Default: Source
- Required: No
- Restrictions: This parameter is not used if DATA-SHARING(NO) is specified.

PRIMARY (SOURCE | YES | NO)

Specifies whether the rebuilt target BSDS for this member will be for the primary of the DB2 data-sharing group. SOURCE specifies that the primary BSDS will remain as the primary. YES specifies that this will be the primary member. NO specifies that this will not be the primary member.

- Default: Source
- Required: No
- Restrictions: This parameter is not used if DATA-SHARING(NO) is specified.

SIMULATE

Specifies that the actions to alter the BSDS are printed as control cards to the DB2 DSNJU003 utility but no modifications are made to the target BSDSs.

- Default: None
- Required: No
- Restrictions: If SIMULATE is not specified, the previous DB2UPDATE must not have been a SIMULATE.
- Short form: SIM

DB2RBLDBSDS step JCL example

This topic contains an example of DB2RBLDBSDS step JCL. Sample JCL can be found in the installation library SCKZJCL in member CKZDRBBS.

For completeness and to illustrate where JCL parameters and DB2 Cloning Tool control statements must match, the following JCL includes sample DB2 Cloning Tool control statements.

The DB2RBLDBSDS step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

      //?????? JOB , 'DB2RBLDBSDS', CLASS=A, MSGCLASS=X
1 //S1      EXEC PGM=CKZ00010, REGION=8M
2 //STEPLIB DD DSN=HLQ?.SCKZLOAD, DISP=SHR
      //      DD DSN=DSNxxx.SDSNLOAD, DISP=SHR
3 //CKZINI DD DSN=HLQ?.SCKZPARM(CKZINI), DISP=SHR
4 //CKZPRINT DD SYSOUT=*
      //SYSUDUMP DD SYSOUT=*
5 //JOURNAL DD DSN=CKZ.JRNL, DISP=OLD
      //CCKZIN DD *
          DB2RBLDBSDS -
6          DB2-MEMBER(DB1T) -
          JOURNAL-DDN(JOURNAL)
      //*

```

1. Execution of DB2 Cloning Tool main program.
2. DB2 Cloning Tool LOAD library must be authorized.
3. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the DB2 Cloning Tool programs.
4. DD for CKZPRINT output.
5. Journal data set used by DB2 Cloning Tool commands. This VSAM data set is used to pass information between DB2 Cloning Tool steps, and to log information derived from the step executions. The journal file is allocated and cataloged in the COPY step and used as input and output by the RENAME step. The specified data set name must match the data set allocated in the COPY step. If multiple DB2 Cloning Tool setups are used for different groups of volumes, take care in specifying the journal data sets for each setup. In the sample JCL, the control statement JOURNAL-DDN(JOURNAL) specifies that a DD with the name JOURNAL is being used rather than a data set name.
6. DB2-MEMBER parameter specifying the member, DB1T, whose BSDS data sets will be rebuilt.

DB2SETLOG

This command is not required. This command is used only for online cloning of a DB2 subsystem using methods other than consistent FlashCopy, SnapShot, or TimeFinder/Clone, or consistent split or break mirror.

DB2SETLOG is used to suspend or resume a source DB2 subsystem, via a DB2 SET LOG command, as part of online cloning a DB2 subsystem. For a DB2 data-sharing group, the SET LOG command must be run on each active member.

For SUSPEND, a SET LOG LOGLOAD(0) command is issued followed by a SET LOG SUSPEND command.

For RESUME, a SET LOG RESUME command is issued.

The userid that runs the DB2SETLOG command must be authorized to connect to the DB2 subsystem and issue the SET LOG commands.

DB2SETLOG command syntax

DB2SETLOG

Required keywords:

DB2-SSID(*db2 ssid*)
 SUSPEND | RESUME

Optional keywords:

SIMULATE

DB2SETLOG command and keyword definitions

Required keywords are described first, followed by optional keywords.

DB2SETLOG

Optional command to suspend or resume a DB2 subsystem.

- Required: No
- Restrictions: This command is only used for cloning of a DB2 subsystem.

DB2-SSID(*db2-ssid*)

This parameter supplies the DB2 SSID of the DB2 subsystem to be suspended or resumed. A group name should not be used for this parameter.

- Default: None
- Required: Yes
- Restrictions: None

SUSPEND

Specifies that a suspend of the DB2 subsystem will be done.

- Default: None
- Required: Yes
- Restrictions: Mutually exclusive with RESUME.

RESUME

Specifies that a resume of the DB2 subsystem will be done.

- Default: None
- Required: No
- Restrictions: Mutually exclusive with RESUME.

SIMULATE

Specifies that the DB2 SET LOG command will not be issued.

- Default: None
- Required: No
- Restrictions: None.
- Short form: SIM

DB2SETLOG step JCL example

|
|
|

There are two DB2SETLOG step JCL examples: one to suspend a DB2 subsystem, and the other to resume a DB2 subsystem. Sample JCL for both can be found in the installation library SCKZJCL in member CKZDSETL.

Suspend a DB2 subsystem

For completeness and to illustrate where JCL parameters and DB2 Cloning Tool control statements must match, the following JCL includes sample DB2 Cloning Tool control statements.

The DB2SETLOG step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

      //??????? JOB , 'DB2SETLOG', CLASS=A, MSGCLASS=X
1 //S1      EXEC PGM=CKZ00010, REGION=6M
2 //STEPLIB DD DSN=HLQ?.SCKZLOAD, DISP=SHR
3 //        DD DSN=DSNxxx.SDSNLOAD, DISP=SHR
4 //CKZINI  DD DSN=HLQ?.SCKZPARAM(CKZINI), DISP=SHR
5 //CKZPRINT DD SYSOUT=*
  //SYSUDUMP DD SYSOUT=*
  //CKZIN   DD *
      DB2SETLOG      -
6      DB2-SSID(DB1S) -
7      SUSPEND
  //*

```

1. Execution of DB2 Cloning Tool main program.
2. DB2 Cloning Tool LOAD library must be authorized.
3. DB2 LOAD library must be authorized.
4. DD for CKZINI, SCKZPARAM member. The CKZINI member of the HLQ?.SCKZPARAM library provides variables to the DB2 Cloning Tool programs.
5. DD for CKZPRINT output.
6. DB2-SSID parameter specifying the DB2 ssid, DB1S, of the DB2 subsystem that will be suspended.
7. SUSPEND parameter specifying that the DB2 subsystem will be suspended.

Resume a DB2 subsystem

For completeness and to illustrate where JCL parameters and DB2 Cloning Tool control statements must match, the following JCL includes sample DB2 Cloning Tool control statements.

The DB2SETLOG step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

      //??????? JOB , 'DB2SETLOG', CLASS=A, MSGCLASS=X
1 //S1      EXEC PGM=CKZ00010, REGION=6M
2 //STEPLIB DD DSN=HLQ?.SCKZLOAD, DISP=SHR
3 //        DD DSN=DSNxxx.SDSNLOAD, DISP=SHR
4 //CKZINI  DD DSN=HLQ?.SCKZPARAM(CKZINI), DISP=SHR
5 //CKZPRINT DD SYSOUT=*
  //SYSUDUMP DD SYSOUT=*
  //CKZIN   DD *
      DB2SETLOG      -
6      DB2-SSID(DB1S) -
7      RESUME
  //*

```

1. Execution of DB2 Cloning Tool main program.
2. DB2 Cloning Tool LOAD library must be authorized.
3. DB2 LOAD library must be authorized.
4. DD for CKZINI, SCKZPARAM member. The CKZINI member of the HLQ?.SCKZPARAM library provides variables to the DB2 Cloning Tool programs.
5. DD for CKZPRINT output.
6. DB2-SSID parameter specifying the DB2 ssid, DB1S, of the DB2 subsystem that will be resumed.
7. RESUME parameter specifying that the DB2 subsystem will be resumed.

DB2SQL

This command is optional and is used only for cloning of a DB2 subsystem. DB2SQL generates and runs the SQL statements necessary to update the DB2 catalog.

The VCATNAME, storage group names, and volumes are updated in the DB2 catalog. The VCATNAME and storage group names come from the DB2 Cloning Tool DB2UPDATE command from the DB2-HLQS and STOGROUPS keywords, and the volumes come from the DB2 Cloning Tool COPY command.

The DB2SQL command can be run only after the DB2UPDATE command has been run.

The userid that runs the DB2SQL command must be defined in the special zparms as either SYSADM or SYSADM2.

To update the DB2 catalog on the target DB2 subsystem, the DB2 Cloning Tool DB2SQL command uses SQL and has a plan and package that must be installed. For more information about installing the plan and package, see the topic “Cloning a DB2 subsystem” on page 80.

The WLM ENVIRONMENT values that DB2 uses for stored procedures and functions can be updated by using the WLM-ENVIRONMENT-MASKS keyword.

| When the DB2SQL command runs, a user-defined index on the DB2 catalog that
| takes an extent before the DB2SQL command completes can create problems.
| User-defined indexes on the DB2 catalog are logically part of the DB2 catalog, but
| they are treated by DB2 as non-system objects. If DB2 must extend a user-defined
| index, it uses the STOGROUP definitions in effect at that time. If the updates to the
| STOGROUP definitions have not been completed, DB2 might produce an error
| when attempting to extend the user-defined index.

For more information about this command, see Chapter 9, “Cloning DB2 subsystems,” on page 95.

| **Note:** DB2 Cloning Tool commands DB2ALTERBSDS, DB2LGRNXCLEAN,
| DB2RBLDBSDS, DB2SQL, DB2UPDATE, DB2UTILXCLEAN, and DB2XCFCLEAN
| must use the same value for DB2-NAME to correctly process for this DB2
| subsystem or data-sharing group.

DB2SQL command syntax

DBSQL

Required keywords:

DB2-SSID(*db2 ssid*)
{ JOURNAL-DSN(*data set name*) | JOURNAL-DDN(*ddname*) }

Optional keywords:

DB2-NAME (*name*)
LISTSQL(Y | N)
SIMULATE
WLM-ENVIRONMENT-MASKS(*mask pairs*)
WLM-ENV-NOT-UPDATED(RC(*nn* | 4))
DATACLAS-MASKS(*mask pairs*)
DATACLAS-NOT-UPDATED(RC(*nn* | 4))

```

MGMTCLAS-MASKS( mask pairs )
MGMTCLAS-NOT-UPDATED( RC( nn | 4 ) )
STORCLAS-MASKS( mask pairs )
STORCLAS-NOT-UPDATED( RC( nn | 4 ) )

```

WLM ENVIRONMENT and SMS class considerations

Oldvalue Syntax

The old value filter mask is used to select the WLM ENVIRONMENT or the SMS class values against which to apply the new value mask. The allowable filter characters are shown in the following table:

Table 55. Filter characters allowed for oldvalue filter masks

Character	Description
*	A single asterisk represents 0 to nn characters of any value.
%	A percent sign represents one non-blank character.
!	An exclamation point represents one national character. @ # \$
<	A lesser-than sign represents one non-numeric character, national symbols included.
>	A greater-than sign represents one numeric character.

Example: Old value filter mask = D%2P* would match WLM ENVIRONMENT value = DB2PENV1

For information about filters and ACS masks, refer to the topic “Use of the U.S.A. EBCDIC code set” on page 22.

Newvalue Syntax

The new value filter mask is used to rename the WLM ENVIRONMENT or SMS class values selected by the old value filter mask. The allowable filter characters are shown in the following table:

Table 56. Filter characters allowed for newvalue filter masks

Character	Description
*	A single asterisk represents 0 to nn characters of any value. The single asterisk may only be used as the last character of the mask.
%	A percent sign represents one non-blank character.

Example: Original WLM ENVIRONMENT value: DB2PWLM1 New value filter mask: TS%T* New WLM ENVIRONMENT value: TS2TENV1

For information about filters and ACS masks, refer to the topic “Use of the U.S.A. EBCDIC code set” on page 22.

DB2SQL command and keyword definitions

Required keywords are described first, followed by optional keywords.

DB2SQL

Optional command to update the DB2 catalog.

- Required: No
- Restrictions: The DB2SQL command can be run only after the DB2UPDATE command has been run.

DB2-SSID(*db2-ssid*)

This parameter supplies the DB2 SSID of the DB2 subsystem to connect to. A group name should not be used for this parameter.

- Default: None
- Required: Yes
- Restrictions: None

**JOURNAL-DSN (*data set name*)
 or JOURNAL-DDN (*ddname*)**

Supplies either the data set name of the DB2 Cloning Tool journal file or the ddname of the DD statement in the JCL that points to the DB2 Cloning Tool journal file.

The journal is used to pass information between DB2 Cloning Tool steps (for example, from the COPY step to the RENAME step). Therefore, as noted in the JCL comments, it must be cataloged in the COPY step when first created, and referenced as OLD in subsequent steps.

Do not delete the data set in the last step, in case restarts or reruns need journal information. It is preferable to delete and replace the data set only at the fresh start of a COPY.

- Default: None
- Required: Yes
- Restrictions: None

DB2-NAME(*name*)

Specifies a name that is the same name as the one used for the prior DB2UPDATE for this DB2 subsystem or data-sharing group.

Note: DB2 Cloning Tool commands DB2ALTERBSDS, DB2LGRNXCLEAN, DB2RBLDBSDS, DB2SQL, DB2UPDATE, DB2UTILXCLEAN, and DB2XCFCLEAN must use the same value for DB2-NAME to correctly process for this DB2 subsystem or data-sharing group.

DB2-NAME is intended to be used when more than one DB2 subsystem or data-sharing group is being cloned from the same DB2 Cloning Tool COPY and RENAME. Name can be 1 - 4 alpha-numeric-national characters.

- Default: None
- Required: No
- Restrictions: None

LISTSQL(*Y | N*)

Specifies that the generated SQL is to be included in the listing.

- Default: N
- Required: No
- Restrictions: None

SIMULATE

Specifies that the generated SQL commands will not be issued.

- Default: None
- Required: No
- Restrictions: None.
- Short form: SIM

WLM-ENVIRONMENT-MASKS (*mask pairs*)

WLM-ENVIRONMENT-MASKS are specified in *oldvalue newvalue* pairs.

WLM-ENVIRONMENT-MASKS are processed in order. The first hit of an old value mask to a source WLM ENVIRONMENT value is the new value mask of that pair that is used to create the target WLM ENVIRONMENT value.

- Default: None
- Required: No
- Restrictions: See the topic “WLM ENVIRONMENT and SMS class considerations” on page 400.

WLM-ENV-NOT-UPDATED (RC(*nn* | 4))

Supplies the return code that is used if there is a WLM ENVIRONMENT value in SYSIBM.SYSROUTINES that is not updated.

- Default: RC(4)
- Required: No
- Restrictions: Used only if WLM-ENVIRONMENT-MASKS is specified.

DATACLAS-MASKS (*mask pairs*)

DATACLAS-MASKS are specified in *oldvalue newvalue* pairs.

DATACLAS-MASKS are processed in order. The first hit of an old value mask to a source DATACLAS value is the new value mask of that pair that is used to create the target DATACLAS value.

- Default: N
- Required: No
- Restrictions: See the topic “WLM ENVIRONMENT and SMS class considerations” on page 400.

DATACLAS-NOT-UPDATED (RC(*nn* | 4))

Supplies the return code that is used if there is a DATACLAS value in SYSIBM.SYSSTOGROUP that is not updated.

- Default: RC(4)
- Required: No
- Restrictions: Used only if DATACLAS-MASKS is specified. Only valid for DB2 Version 9.1.

MGMTCLAS-MASKS (*mask pairs*)

MGMTCLAS-MASKS are specified in *oldvalue newvalue* pairs.

MGMTCLAS-MASKS are processed in order. The first hit of an old value mask to a source MGMTCLAS value is the new value mask of that pair that is used to create the target MGMTCLAS value.

- Default: None
- Required: No
- Restrictions: See the topic “WLM ENVIRONMENT and SMS class considerations” on page 400.

MGMTCLAS-NOT-UPDATED (RC(*nn* | 4))

Supplies the return code that is used if there is a MGMTCLAS value in SYSIBM.SYSSTOGROUP that is not updated.

- Default: RC(4)
- Required: No
- Restrictions: Used only if MGMTCLAS-MASKS is specified. Only valid for DB2 Version 9.1.

STORCLAS-MASKS (*mask pairs*)

STORCLAS-MASKS are specified in *oldvalue newvalue* pairs.

STORCLAS-MASKS are processed in order. The first hit of an old value mask to a source STORCLAS value is the new value mask of that pair that is used to create the target STORCLAS value.

- Default: None
- Required: No
- Restrictions: See the topic “WLM ENVIRONMENT and SMS class considerations” on page 400.

STORCLAS-NOT-UPDATED (RC(*nn* | 4))

Supplies the return code that is used if there is a STORCLAS value in SYSIBM.

- Default: RC(4)
- Required: No
- Restrictions: Used only if STORCLAS-MASKS is specified. Only valid for DB2 Version 9.1.

DB2SQL step JCL example

This topic contains an example of DB2SQL step JCL. Sample JCL can be found in the installation library SCKZJCL in member CKZDSQL.

For completeness and to illustrate where JCL parameters and DB2 Cloning Tool control statements must match, the following JCL includes sample DB2 Cloning Tool control statements.

The DB2SQL step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

1 //??????? JOB , 'DB2SQL', CLASS=A, MSGCLASS=X
2 //S1      EXEC PGM=CKZ00010, REGION=8M
3 //STEPLIB DD DSN=HLQ?.SCKZLOAD, DISP=SHR
4 //        DD DSN=DSNxxx.SDSNLOAD, DISP=SHR
5 //CKZINI  DD DSN=HLQ?.SCKZPARAM(CKZINI), DISP=SHR
6 //CKZPRINT DD SYSOUT=*
7 //SYSUDUMP DD SYSOUT=*
8 //JOURNAL DD DSN=CKZ.JRNL, DISP=OLD
9 //CKZIN   DD *
10          DB2SQL           -
11          DB2-SSID(DB1T)   -
12          LISTSQL(Y)       -
13          JOURNAL-DDN(JOURNAL)
14 //*

```

1. Execution of DB2 Cloning Tool main program.
2. DB2 Cloning Tool LOAD library must be authorized.
3. DB2 LOAD library must be authorized.
4. DD for CKZINI, SCKZPARAM member. The CKZINI member of the HLQ?.SCKZPARAM library provides variables to the DB2 Cloning Tool programs.
5. DD for CKZPRINT output.
6. Journal data set used by DB2 Cloning Tool commands. This VSAM data set is used to pass information between DB2 Cloning Tool steps, and to log information derived from the step executions. The journal file is allocated and cataloged in the COPY step, and used as input and output by the RENAME step. The specified data set name must match the data set allocated in the COPY step. If multiple DB2 Cloning Tool processes are created for different

applications, each must use a unique journal data set. In the sample JCL, the control statement JOURNAL-DDN(JOURNAL) specifies that a DD with the name JOURNAL is being used, rather than a data set name.

7. DB2-SSID parameter specifying DB2 ssid, DB1T, that will be connected to.
8. LISTSQL parameter specifying the generated SQL statements should be included in the listing.

DB2START

This command is not required. This command is used only for cloning of a DB2 subsystem.

DB2START is used to start a DB2 subsystem, via a z/OS START DB2 command, as part of cloning a DB2 subsystem. The DB2 subsystem can be started in a special maintenance mode, maintenance mode, or normal mode. The SPECIAL, MAINT, and NORMAL keywords are used to specify the DB2 subsystem start mode.

Using the SPECIAL keyword, the DB2 subsystem will be started in maintenance mode with a special dsnzparm module. This special dsnzparm module is checked to verify that it has the correct settings: DEFER,ALL and the DB2 catalog updatable attribute. (See the topic “Cloning a DB2 subsystem” on page 80 for how to set up the special dsnzparm module.)

After the START DB2 command is issued, DB2START waits for the DB2 subsystem to complete its start-up. The WAIT keyword is used to specify how long this wait is and the return code to be used when the wait time is exceeded. When the LIGHT keyword is specified, DB2START will wait until the DB2 subsystem stops after it starts up.

Using the REPLY-TO-RESTART-WTOR(Y) keyword, DB2START will automatically reply Y to the DB2 DSNJ245I or DSNJ246I restart WTOR message. These WTOR messages can occur when a DB2ALTERBSDS command using the COLD-START or SLB-START keywords was run prior to DB2START.

Using the STOP-WAITING-IF-DSNR020I(Y ,RC(n)) keyword, DB2START will stop waiting if a DSNR020I WTOR is issued and will provide the return code specified in the keyword (or the default return code if not specified). The DSNR020I WTOR message can occur when cloning a data sharing group from a DB2 system level backup (DB2 BACKUP SYSTEM). When the target DB2 members are started, they may receive the DSNR020I WTOR if a conditional restart record was created in each member to have its log truncated to the BACKUP SYSTEM data complete LRSN. The conditional restart record can be created by the DB2ALTERBSDS command with the SLB-START keyword. To allow for better automation of the cloning process in this situation, use the STOP-WAITING-IF-DSNR020I(Y,RC(n)) keyword. If the start jobs are being run serially by a job scheduler, it is recommended that the DB2START of all but the last member should include the STOP-WAITING-IF-DSNR020I keyword with a value of Y. This will allow members to be started concurrently. After the last member has been started, a DB2START for the first member should be reissued using the WAITONLY and DB2-ALREADY-RUNNING(RC(0)) keywords. Completion of this DB2START with a return code of 4 or less indicates that the first member is up and ready for the remainder of the DB2 conditioning jobs. For an example of how to use the STOP-WAITING-IF-DSNR020I keyword, refer to the step for running DB2 conditioning commands (Step 8) in “DB2 subsystem cloning from a DB2 BACKUP SYSTEM backup” on page 538.

The userid that runs the DB2START command must be authorized to issue z/OS START DB2 commands and to connect to the DB2 subsystem. When the SPECIAL keyword is used, the userid must also be defined in the special dsnzparm as either SYSADM or SYSADM2. When the MAINT keyword is used, the userid must also be defined in the dsnzparm as either SYSADM or SYSADM2.

A target DB2 subsystem will use the same buffer pool specifications as its corresponding source DB2 subsystem. If the buffer pool definitions in the source DB2 subsystem are large, care should be taken that sufficient real and auxiliary storage exists to support the size of the buffer pools in the target DB2 subsystem until ALTER BUFFERPOOL commands can be issued.

If the DB2 system is data sharing and the DDF LOCATION is being changed, when the target DB2 systems are started they may issue the message:

```
DSNJ707E LOCATION NAME location-name IN BSDS DOES NOT MATCH THE LOCATION
NAME location-name ASSOCIATED WITH THE DATA SHARING GROUP.
```

This message does not indicate a problem. The message is issued due to the way DB2 rebuilds the SCA when the target DB2 system is initially started.

If there are DDF transactions active during the cloning, when the target DB2 systems are started they may issue the message:

```
DSNL034E DDF CANNOT BE STARTED BECAUSE OF BSDS INCONSISTENCIES
```

To resolve this condition and allow DDF to start, the following DB2 command must be issued on the target DB2 system:

```
-RESET INDOUBT LUNAME(*) FORCE
```

This DB2 command should not be issued until after the DB2FIX command with the DATABASES(APPLICATION) keyword has been run.

DB2START command syntax

DB2START

Required keywords:

```
DB2-SSID( db2 ssid )
NORMAL | MAINT | SPECIAL
```

Optional keywords:

```
DB2-ALREADY-RUNNING( RC( nn | 8 ) )
DSNZPARM( zparm-name )
LIGHT
| MSTR-DETECT-WAIT( [ nnn | 1 ] [, RC( nn | 8 ) ] )
| REPLY-TO-RESTART-WTOR( Y | N )
| SIMULATE
| STOP-WAITING-IF-DB2-STOPS( N )
| | STOP-WAITING-IF-DB2-STOPS( [ Y ] [, RC( nn | 8 ) ] )
| STOP-WAITING-IF-DSNR020I( N )
| | STOP-WAITING-IF-DSNR020I( Y [, RC( nn | 3 ) ] )
| WAIT( [ nnn | 5 ] [,RC( nn | 8 ) ] )
| WAITONLY
```

DB2START command and keyword definitions

Required keywords are described first, followed by optional keywords.

DB2START

Optional command to start a DB2 subsystem.

- Required: No
- Restrictions: This command is only used for cloning of a DB2 subsystem.

DB2-SSID(*db2-ssid*)

This parameter supplies the DB2 SSID of the DB2 subsystem to start. A group name should not be used for this parameter.

- Default: None
- Required: Yes
- Restrictions: None

NORMAL

Specifies that a normal start of the DB2 subsystem will be done.

- Default: None
- Required: Yes
- Restrictions: Mutually exclusive with MAINT and SPECIAL.

SPECIAL

Specifies that a special start of the DB2 subsystem will be done.

- Default: None
- Required: Yes
- Restrictions: Mutually exclusive with NORMAL and MAINT. If SPECIAL is specified, DSNZPARM must also be specified.

MAINT

Specifies that a MAINT start of the DB2 subsystem will be done.

- Default: None
- Required: Yes
- Restrictions: Mutually exclusive with NORMAL and SPECIAL

DB2-ALREADY-RUNNING(RC(*nn* | 8))

This parameter supplies the return code that will be used if the DB2 subsystem is already running.

- Default: RC(8)
- Required: No
- Restrictions: None

DSNZPARM(*zparm-name*)

This parameter supplies the dsnzparm module name that will be used in the start DB2 command.

- Default: None
- Required: Only required if SPECIAL is specified.
- Restrictions: None

LIGHT

Specifies that LIGHT(YES) will be used in the start command.

- Default: None
- Required: No
- Restrictions: Mutually exclusive with STOP-WAITING-IF-DB2-STOPS.

MSTR-DETECT-WAIT([*mmm* | 1] [, RC(*rr* | 8)])

mmm specifies the maximum time in minutes that DB2START should wait for the DB2 xxxxMSTR address space to start executing. If the specified

| time limit expires before the DB2 xxxxMSTR address space has started
| executing, DB2START will terminate with a return code of *rr*.

- | • Default: MSTR-DETECT-WAIT(1,RC(8))
- | • Required: No
- | • Restrictions: None

REPLY-TO-RESTART-WTOR(Y | N)

Specifies whether DB2START should make a Y reply to the DB2 DSNJ245I or DSNJ246I restart WTOR message.

This option addresses the situation where DB2 puts out a WTOR during its start up and it is desired to automatically make a Y reply to it. The DB2 restart WTOR needing the reply is a result of a conditional restart record that is causing either a cold start or a truncation of the DB2 log. This type of conditional restart record can be created by the DB2ALTERBSDS command using the COLD-START or SLB-START keywords.

The full text of the DB2 restart WTOR messages addressed by this keyword:

```
DSNJ245I CONDITIONAL RESTART RECORD INDICATES TRUNCATION AT RBA rrr.  
REPLY Y TO CONTINUE, N TO CANCEL  
DSNJ246I CONDITIONAL RESTART RECORD INDICATES COLD START AT RBA rrr.  
REPLY Y TO CONTINUE, N TO CANCEL
```

- Default: N
- Required: No
- Restrictions: None

SIMULATE

Specifies that the start command will not be issued.

- Default: None
- Required: No
- Restrictions: None.
- Short form: SIM

STOP-WAITING-IF-DB2-STOPS(N)

| or STOP-WAITING-IF-DB2-STOPS([Y] [, RC(*nn* | 8)])

| Specifies whether DB2START should stop waiting for the DB2 subsystem
| to start if the DB2 subsystem stops prematurely.

| N specifies that DB2START should not stop waiting if the DB2 subsystem
| stops prematurely. The WAIT keyword will control how long DB2START
| will wait and the return code that is issued when the wait time has been
| exceeded. Y specifies that DB2START should stop waiting if the DB2
| subsystem stops prematurely and DB2START will terminate with a return
| code of *nn*.

- | • Default: STOP-WAITING-IF-DB2-STOPS (Y,RC(8))
- | • Required: No
- | • Restrictions: Mutually exclusive with LIGHT.

STOP-WAITING-IF-DSNR020I(N)

| or STOP-WAITING-IF-DSNR020I(Y [, RC(*nn* | 3)])

| Specifies whether DB2START should stop waiting for the DB2 subsystem
| to start if the DSNR020I WTOR is received.

| N specifies that DB2START should continue to wait if the DSNR020I
| WTOR is received. The WAIT keyword will control how long DB2START

will wait and the return code that is issued when the wait time has been exceeded. Y specifies that DB2START should stop waiting if the DSNR020I WTOR is received. DB2START will terminate with a return code of *nn*. The full text of the DB2 restart WTOR message addressed by this keyword is DSNR020I START MEMBER *member*, OR REPLY 'NO' OR 'QUIESCED'.

- Default: STOP-WAITING-IF-DSNR020I(N)
- Required: No
- Restrictions: None

WAIT([*nnn* | 5] [,RC(*rr* | 8)])

nnn specifies the maximum time in minutes that DB2START should wait for the DB2 subsystem to start. If the specified time limit expires before the DB2 subsystem has started, DB2START will terminate with a return code of *rr*.

- Default: WAIT(5,RC(8))
- Required: No
- Restrictions: None

WAITONLY

Specifies that the start command will not be issued but the wait for the DB2 subsystem to start will be done.

- Default: None
- Required: No
- Restrictions: None

DB2START step JCL example

This topic contains an example of DB2START step JCL. Sample JCL can be found in the installation library SCKZJCL in member CKZDSTA.

For completeness and to illustrate where JCL parameters and DB2 Cloning Tool control statements must match, the following JCL includes sample DB2 Cloning Tool control statements.

The DB2START step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

      //??????? JOB , 'DB2START', CLASS=A, MSGCLASS=X
1 //S1      EXEC PGM=CKZ00010, REGION=6M
2 //STEPLIB DD DSN=HLQ?.SCKZLOAD, DISP=SHR
3 //        DD DSN=DSNxxx.SDSNEXIT, DISP=SHR
4 //        DD DSN=DSNxxx.SDSNLOAD, DISP=SHR
5 //CKZINI  DD DSN=HLQ?.SCKZPARAM(CKZINI), DISP=SHR
6 //CKZPRINT DD SYSOUT=*
      //SYSUDUMP DD SYSOUT=*
      //CKZIN  DD *
          DB2START          -
7          DB2-SSID(DB1T)  -
8          SPECIAL         -
9          DSNZPARAM(DB1TSPEC)
      //*
```

1. Execution of DB2 Cloning Tool main program.
2. DB2 Cloning Tool LOAD library must be authorized.
3. DB2 EXIT library, must be authorized.
4. DB2 LOAD library must be authorized.

5. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the DB2 Cloning Tool programs.
6. DD for CKZPRINT output.
7. DB2-SSID parameter specifying the DB2 ssid, DB1T, of the DB2 subsystem that will be started.
8. SPECIAL parameter specifying that the DB2 subsystem will be started in maintenance mode with the special dsnzparm DB1TSPEC.
9. DSNZPARM parameter specifying the module name of the special dsnzparm.

DB2STOP

This command is not required. This command is used only for cloning of a DB2 subsystem.

DB2STOP is used to stop a DB2 subsystem, via the DB2 STOP DB2 command, as part of cloning a DB2 subsystem.

After the STOP DB2 command is issued, DB2STOP waits for the DB2 subsystem to terminate. The WAIT keyword is used to specify how long this wait is and the return code to be used when the wait time is exceeded.

The userid that runs the DB2STOP command must be authorized to connect to the DB2 subsystem and issue the STOP DB2 command. When the DB2 subsystem is running in maintenance mode, the userid must be defined in the dsnzparm as either SYSADM or SYSADM2.

DB2STOP command syntax

DB2STOP

Required keywords:

DB2-SSID(*db2 ssid*)

Optional keywords:

CASTOUT(NO | YES)
 DB2-ALREADY-STOPPED(RC(*nn* | 8))
 MODE(FORCE | QUIESCE)
 SIMULATE
 WAIT([*nnn* | 5] [,RC(*nn* | 8)])
 WAITONLY

DB2STOP command and keyword definitions

Required keywords are described first, followed by optional keywords.

DB2STOP

Optional command to stop a DB2 subsystem.

- Required: No
- Restrictions: This command is only used for cloning of a DB2 subsystem.

DB2-SSID(*db2-ssid*)

This parameter supplies the DB2 SSID of the DB2 subsystem to stop. A group name should not be used for this parameter.

- Default: None
- Required: Yes

- Restrictions: None

CASTOUT(NO | YES)

This parameter supplies the CASTOUT value that will be used in the stop DB2 command.

- Default: YES
- Required: No
- Restrictions: None

DB2-ALREADY-STOPPED(RC(*nn* | 8))

This parameter supplies the return code that will be used if the DB2 subsystem is already stopped.

- Default: RC(8)
- Required: No
- Restrictions: None

MODE(FORCE | QUIESCE)

This parameter supplies the MODE value that will be used in the stop DB2 command. See the DB2 Command Reference for the meanings of the FORCE and QUIESCE keywords and their values.

- Default: QUIESCE
- Required: No
- Restrictions: None

SIMULATE

Specifies that the stop command will not be issued.

- Default: None
- Required: No
- Restrictions: None.
- Short form: SIM

WAIT([*nnn* | 5] [,RC(*rr* | 8)])

nnn specifies the maximum time in minutes that DB2STOP should wait for the DB2 subsystem to stop. If the specified time limit expires before the DB2 subsystem has stopped, DB2STOP will terminate with a return code of *rr*

- Default: WAIT(5,RC(8))
- Required: No
- Restrictions: None

WAITONLY

Specifies that the stop command will not be issued but the wait for the DB2 subsystem to stop will be done.

- Default: None
- Required: No
- Restrictions: None

DB2STOP step JCL example

| This topic contains an example of DB2STOP step JCL. Sample JCL can be found in
| the installation library SCKZJCL in member CKZDSTO.

For completeness and to illustrate where JCL parameters and DB2 Cloning Tool control statements must match, the following JCL includes sample DB2 Cloning Tool control statements.

The DB2STOP step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```
    //??????? JOB , 'DB2STOP', CLASS=A, MSGCLASS=X
1 //S1      EXEC PGM=CKZ00010, REGION=6M
2 //STEPLIB DD DSN=HLQ?.SCKZLOAD, DISP=SHR
3 //        DD DSN=DSNxxx.SCKZLOAD, DISP=SHR
4 //CKZINI  DD DSN=HLQ?.SCKZPARAM(CKZINI), DISP=SHR
5 //CKZPRINT DD SYSOUT=*
  //SYSUDUMP DD SYSOUT=*
  //CKZIN   DD *
    DB2STOP -
6    DB2-SSID(DB1T)
  //*
```

1. Execution of DB2 Cloning Tool main program.
2. DB2 Cloning Tool LOAD library must be authorized.
3. DB2 LOAD library must be authorized.
4. DD for CKZINI, SCKZPARAM member. The CKZINI member of the HLQ?.SCKZPARAM library provides variables to the DB2 Cloning Tool programs.
5. DD for CKZPRINT output.
6. DB2-SSID parameter specifying the DB2 ssid, DB1T, of the DB2 subsystem that will be stopped.

DB2UPDATE

This command is optional. If a DB2 subsystem is being cloned, with the intent of a second DB2 subsystem accessing the renamed data sets, DB2UPDATE might be used to make some of the changes within DB2 to reflect the renamed data sets.

DB2UPDATE updates the DB2 directory and the DB2 bootstrap data sets (BSDSs). For data sharing, DB2UPDATE can deallocate the target DB2 XCF structures and remove the target DB2 XCF member information.

Active log data sets in the BSDSs are modified based on the rename masks supplied in the RENAME command. ARCHIVE data sets are modified only if the optional ARCHIVE keyword is specified.

The DB2 directory information that is updated resides in table space DBD01 in database DSNDB01. For DB2 10 NFM, this information was moved to LOB table space SYSDBDXA in database DSNDB01. In the DB2UPDATE JCL, a DD statement that is named DBD01 points to the table space DBD01 in database DSNDB01. For DB2 10 NFM, DB2UPDATE can detect that the information to be updated was moved from the table space DBD01 and will dynamically allocate LOB table space SYSDBDXA, based on the data set name that is allocated to the DBD01 DD statement, and update it. Another option for DB2 10 NFM is to add to the DB2UPDATE JCL a DD statement with a name of SYSDBDXA that points to LOB table space SYSDBDXA in database DSNDB01.

Note: If the archive logs are on tape, you can run the DB2 Cloning Tool DB2ALTERBSDS command to delete the source archive log file names from the target BSDS data sets. This optional command can run any time after DB2 Cloning

Tool DB2UPDATE completes and before or after the target DB2 subsystem is up. You can leave the original names in the target BSDSs if you prefer.

The target DB2 subsystem cannot be started until the DB2 Cloning Tool COPY, RENAME, and DB2UPDATE commands complete. The target DB2 subsystem must be stopped before the source subsystem is cloned again (next DB2 Cloning Tool COPY command).

DB2UPDATE can be run only when the previous RENAME was not a SIMULATE.

For data sharing, DB2UPDATE must be run for each member of the data sharing group. The first DB2UPDATE would be run for the primary member and would update the DB2 directory and that member's BSDSs. Then, DB2UPDATE would be run for each of the other members in the data sharing group and would include the BSDSONLY keyword, which would update only that member's BSDSs.

If the DB2 system is data sharing and the DDF LOCATION is being changed, when the target DB2 systems are started, they might issue the message

```
DSNJ707E LOCATION NAME location-name IN BSDS DOES NOT MATCH THE LOCATION
NAME location-name ASSOCIATED WITH THE DATA SHARING GROUP.
```

This message does not indicate a problem. The message is issued due to the way DB2 rebuilds the SCA when the target DB2 system is initially started.

Note: DB2 Cloning Tool commands DB2ALTERBSDS, DB2LGRNXCLEAN, DB2RBLDBSDS, DB2SQL, DB2UPDATE, DB2UTILXCLEAN, and DB2XCFCLEAN must use the same value for DB2-NAME to correctly process for this DB2 subsystem or data-sharing group.

For more information about offline or online DB2 cloning, see the topic Chapter 9, "Cloning DB2 subsystems," on page 95.

DB2UPDATE command syntax

DB2UPDATE

Required keywords:

```
DB2-HLQS( sourcealias1, targetalias1, ... )
{ JOURNAL-DSN( data set name ) | JOURNAL-DDN( ddname ) }
```

Required keywords when the source DB2 is data sharing:

```
DB2-GROUP( sourcegroup, targetgroup )
DB2-MEMBERS( sourcemember1, targetmember1, ... )
```

Optional keywords:

```
ARCHIVE
BSDSONLY | DBD01ONLY
DB2-NAME( name )
DB2-XCFCLEAN( Y | N )
DDF(
  ALIAS( alias1, alias2, ... | NOALIAS )
  GENERIC( genericluname | NGENERIC )
  GRPIP4( ip address | NGRPIP4 )
  GRPIP6( ip address | NGRPIP6 )
  IPNAME( ipname | NOIPNAME )
  IPV4( ip address | NOIPV4 )
  IPV6( ip address ) | NOIPV6 )
```



```

LOCATION( location )
LUNAME( luname | NOLUNAME )
PASSWORD( password | NOPASSWD )
PORT( port )
RESPORT( resport )
SECPORT( secpport )
)
DDF-NOT-UPDATED( RC( nn | 4 ) )
HLQ-NOT-UPDATED( RC( nn | 4 ) )
SIMULATE
STOGROUPS( sourcestorgroup1, targetstorgroup1, ... )
SYSVALUE-DDN( ddname )

```

DB2UPDATE command and keyword definitions

Required keywords are described first, followed by optional keywords.

DB2UPDATE

Optional command to update DB2 to reflect renamed data sets.

- Required: No
- Restrictions: DB2UPDATE can be run only when the previous RENAME was not a SIMULATE.

DB2-HLQS (*sourcealias1, targetalias1, ...*)

Specify the source and target DB2 data set high-level qualifiers. (1-8 character aliases).

- Default: None
- Required: Yes
- Restrictions: None
- Short form: DHLQS

DB2-GROUP (*sourcegroup, targetgroup*)

Specify the source and target DB2 group names.

- Default: None
- Required: This keyword is required when the source DB2 is data sharing.
- Restrictions: None
- Short form: DGRP

DB2-MEMBERS (*sourcemember1, targetmember1, ...*)

Specify the source and target DB2 member names. Each of the member names must be unique.

- Default: None
- Required: This keyword is required when the source DB2 is data sharing.
- Restrictions: None
- Short form: DMEMS

JOURNAL-DSN (*data set name*)

or JOURNAL-DDN (*ddname*)

This parameter supplies either a data set name of the DB2 Cloning Tool journal file or a ddname assumed via the JCL to point at a journal data set. The journal data set for the RENAME step must be the same data set specified for the COPY step.

If multiple DB2 Cloning Tool setups are used for different volume groups, DO NOT use the same journal data set. Each DB2 Cloning Tool "application" needs a different journal data set.

The journal is used to pass information between DB2 Cloning Tool steps (for example, from the COPY step to the RENAME step). Therefore, as noted in the JCL comments, it must be cataloged in the COPY step when first created, and referenced as OLD in subsequent steps.

In case restarts or reruns need journal information, do not delete the data set in the last step. It is preferable to delete and replace the data set only at the fresh start of a COPY.

- Default: None
- Required: Yes
- Restrictions: None

ARCHIVE

If the DB2 archive logs are on the source DASD volumes, and are copied to the target volumes, this keyword changes the data set names and VOLSERS of the DB2 archive logs in the BSDSs to the target values.

- Default: None
- Required: No
- Restrictions: None

BSDSONLY

Indicates that only the DB2 member's BSDS will be updated. No access will be attempted to the DB2 directory table space DBD01.

- Default: None
- Required: No
- Restrictions: Mutually exclusive with DBD01ONLY. If BSDSONLY is specified, DB2-GROUP and DB2-MEMBERS must also be specified.

DBD01ONLY

Indicates that only the DB2 directory table space DBD01 will be updated. No access will be attempted to the DB2 subsystem's BSDS.

- Default: None
- Required: No
- Restrictions: Mutually exclusive with BSDSONLY.

DB2-NAME(*name*)

Specifies an arbitrary name that identifies this DB2 subsystem or data-sharing group. The same name should be used for all members of a data-sharing group.

Note: DB2 Cloning Tool commands DB2ALTERBSDS, DB2LGRNXCLEAN, DB2RBLDBSDS, DB2SQL, DB2UPDATE, DB2UTILXCLEAN, and DB2XCFCLEAN must use the same value for DB2-NAME to correctly process for this DB2 subsystem or data-sharing group.

DB2-NAME is intended to be used when more than one DB2 subsystem or data-sharing group is being cloned from the same DB2 Cloning Tool COPY and RENAME. Name can be 1 - 4 alpha-numeric-national characters.

- Default: None
- Required: No
- Restrictions: None

DB2-XCFCLEAN(Y | N)

Specifies that DB2 Cloning Tool should clean up the target DB2 data-sharing group XCF structures and group members. If not specified in the control statements, the default is obtained from the CKZINI member.

- Default: Y
- Required: No
- Restrictions: Used only if DB2-GROUP is specified.
- Short form: DXCFCLN

DDF(

ALIAS(*alias1, alias2, ...* | **NOALIAS**)

GENERIC(*genericluname* | **NGENERIC**)

GRPIPV4(*ip address* | **NGRPIPV4**)

GRPIPV6(*ip address* | **NGRPIPV6**)

IPNAME(*ipname* | **NOIPNAME**)

IPV4(*ip address* | **NOIPV4**)

IPV6(*ip address*) | **NOIPV6**)

LOCATION(*location*)

LUNAME(*luname* | **NOLUNAME**)

PASSWORD(*password* | **NOPASSWD**)

PORT(*port*)

RESPORT(*resport*)

SECPORT(*secpport*)

-) Indicates that DB2UPDATE should modify the BSDS LBSC record. The record must exist on the source system; DB2 Cloning Tool does not add a new record.

ALIAS specifies 1–8 alias names. An alias name is of the form: *location[:port][:secpport]* where *location* is 1–16 alphanumeric characters; first position must be alpha, and *port* and *secpport* are 1–5 numeric characters. NOALIAS nullifies the field.

GENERIC is 1–8 alpha-numeric-national characters; first position must be alpha or national. NGENERIC nullifies the field.

GRPIPV4 is a dotted decimal form IP address. NGRPIPV4 nullifies the field.

GRPIPV6 is a colon hexadecimal form IP address. NGRPIPV6 nullifies the field.

IPNAME is 1–8 alpha-numeric-national characters; first position must be alpha or national. NOIPNAME nullifies the field.

IPV4 is a dotted decimal form IP address. NOIPV4 nullifies the field.

IPV6 is a colon hexadecimal form IP address. NOIPV6 nullifies the field.

LOCATION is 1–16 alphanumeric characters; first position must be alpha.

LUNAME is 1–8 alpha-numeric-national characters; first position must be alpha or national. NOLUNAME nullifies the field.

PASSWORD is 1–8 alpha-numeric-national characters; first position must be alpha or national. NOPASSWD nullifies the field.

PORT is 1–5 numeric characters.

RESPORT is 1–5 numeric characters.

SECPORT is 1–5 numeric characters. 0 nullifies the field.

- Default: None
- Required: No
- Restrictions:
 - The LBSC record must exist in the target BSDS.

- ALIAS is only valid for DB2 Version 8 or above. The secport option of ALIAS is only valid for DB2 Version 9.1.
- GRPIPv4, GRPIPv6, IPNAME, IPV4, IPV6, NOLUNAME, and SECPort are only valid for DB2 Version 9.1.
- If GRPIPv4 is specified, IPV4 must also be specified.
- If GRPIPv4 is specified, IPV4 must also be specified.
- If IPV4(NOIPV4) is specified, GRPIPv4(NGRPIPv4) must also be specified.
- If IPV6(NOIPV6) is specified, GRPIPv6(NGRPIPv6) must also be specified.

DDF-NOT-UPDATED(RC(*nn* | 4))

Supplies the return code that is used if there is a DDF record in the BSDS and the DDF keyword was not specified.

- Default: RC(4)
- Required: No
- Restrictions: None

HLQ-NOT-UPDATED(RC(*nn* | 4))

Supplies the return code that is used if the VSAM catalog name in the BSDS is not updated or a VCAT in the DB2 directory table space DBD01 is not updated.

If you do not change the VSAM catalog name in the BSDS, or a VCAT in the DB2 directory table space DBD01, the target DB2 system might access table and index spaces on the source DB2 system.

- Default: RC(4)
- Required: No
- Restrictions: None

SIMULATE

Specifies that BSDS records to be changed are printed with the target values, but no modifications are made to the target BSDSs; DB2 directory records to be changed indicate the number of fields that are updated, but no modifications are made to the target DB2 directory.

- Default: None
- Required: No
- Restrictions: None.
- Short form: SIM

STOGROUPS(*sourcestorgroup1*, *targetstorgroup1*, ...)

Specifies the source and target storage groups that are defined to the DB2 subsystem. Each target storage group must be unique. For DB2 Version 8, in any pair, the source and the target storage group names must be the same length and a storage group name can be a maximum of 30 characters.

- Default: None
- Required: No
- Restrictions: For DB2 Version 8 and above, in any pair, the source and the target storage group names must be the same length and a storage group name can be a maximum of 30 characters.

SYSVALUE-DDN(*ddname*)

This parameter specifies the DD name that points to a file where the DB2

VCAT pairing (DB2-HLQS) information will be written. SYSVALUE-DDN is intended to be used in a RESTORE SYSTEM utility that uses the LOGONLY SWITCH VCAT SYSVALUEDDN(*ddname*) parameters as input to the SYSVALUEDDN *ddname*.

- Default: None
- Required: Yes
- Restrictions: None

DB2UPDATE step JCL example

Three DB2UPDATE step JCL examples are included: for a non-data sharing environment, for a data-sharing environment, and for CKZDUPD2. Sample JCL for these examples can be found in the installation library SCKZJCL in members CKZDUPD and CKZDUPD2.

DB2UPDATE Step JCL Example – Non-data sharing environment

For completeness and to illustrate where JCL parameters and DB2 Cloning Tool control statements must match, the following JCL includes sample DB2 Cloning Tool control statements.

The DB2UPDATE step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

1 //??????? JOB , 'DB2UPDATE', CLASS=A, MSGCLASS=X
2 //S1      EXEC PGM=CKZ00010, REGION=8M
3 //STEPLIB DD DSN=HLQ?.SCKZLOAD, DISP=SHR
4 //CKZINI  DD DSN=HLQ?.SCKZPARM(CKZINI), DISP=SHR
5 //CKZPRINT DD SYSOUT=*
6 //SYSUDUMP DD SYSOUT=*
7 //JOURNAL DD DSN=CKZ.JRNL, DISP=OLD
8 //*****
9 //* THE DB2 target MUST BE CHANGED TO MATCH THE HIGH LEVEL QUALIFIER
10 //* FOR THE TARGET SYSTEM
11 //*****
12 //BDS01   DD DISP=OLD, DSN=target.BSDS01
13 //BDS02   DD DISP=OLD, DSN=target.BSDS02
14 //DBD01   DD DISP=OLD,
15 //          DSN=target.DSNDBC.DSNDB01.DBD01.I0001.A001
16 //CKZIN   DD *
17 DB2UPDATE
18 JOURNAL-DDN(JOURNAL)
19 DB2-HLQS(SRCDB21, TGTDB21, SRCDB22, TGTDB22)
20 //*
```

1. Execution of DB2 Cloning Tool main program.
2. DB2 Cloning Tool LOAD library must be authorized.
3. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the DB2 Cloning Tool programs.
4. DD for CKZPRINT output.
5. Journal data set used by DB2 Cloning Tool commands. This VSAM data set is used to pass information between DB2 Cloning Tool steps and logs information derived from the step executions. The journal file is allocated and cataloged in the COPY step, and used as input and output by the RENAME step, and input by the DB2UPDATE step. The specified data set name must match the data set allocated in the COPY step. If multiple DB2 Cloning Tool processes are created for different applications, each must use a unique journal

data set. In the sample JCL, the control statement JOURNAL-DDN(JOURNAL) specifies that a DD with the name JOURNAL be used, rather than a data set name.

6. DD names 'BSDS01' and 'BSDS02' pointing to names of target BSDSs.
7. DD name 'DBD01' pointing to name of target 'DSNDBC.DSNDB01.DBD01' data set.
8. CKZIN DD control statement input.
9. DB2 Cloning Tool DB2UPDATE command. This command may be specified in the same execution with COPY and RENAME if desired. When multiple commands are specified in a single execution, if any command completes with a return code GE 8, the remaining commands are flushed without executing.
10. DB2-HLQS parameter specifying that the source and target high level qualifier pairs of SRCDB21,TGTDB21 and SRCDB22,TGTDB22.

DB2UPDATE step JCL example – Data-sharing environment

For completeness and to illustrate where JCL parameters and DB2 Cloning Tool control statements must match, the following JCL includes sample DB2 Cloning Tool control statements.

The DB2UPDATE step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

1 //??????? JOB , 'DB2UPDATE', CLASS=A, MSGCLASS=X
2 //S1 EXEC PGM=CKZ00010, REGION=6M
3 //STEPLIB DD DSN=HLQ?.SCKZLOAD, DISP=SHR
4 //CKZINI DD DSN=HLQ?.SCKZPARM(CKZINI), DISP=SHR
5 //CKZPRINT DD SYSOUT=*
6 //SYSUDUMP DD SYSOUT=*
7 //JOURNAL DD DSN=CKZ.JRNL, DISP=OLD
8 //*****
9 //* THE DB2 target MUST BE CHANGED TO MATCH THE HIGH LEVEL QUALIFIER
10 //* FOR THE TARGET SYSTEM
11 //*****
12 //BSDS01 DD DISP=OLD, DSN=target.BSDS01
13 //BSDS02 DD DISP=OLD, DSN=target.BSDS02
14 //DBD01 DD DISP=OLD,
15 // DSN=target.DSNDBC.DSNDB01.DBD01.I0001.A001
16 //CKZIN DD *
17 DB2UPDATE -
18 JOURNAL-DDN(JOURNAL) -
19 DB2-HLQS(SRCDB21,TGTDB21,SRCDB22,TGTDB22) -
20 DB2-GROUP(DBGS,DBGT) -
21 DB2-MEMBERS(DB1S,DB1T,
22 DB2S,DB2T) -
23 DDF(LUNAME(DB1TLU) PORT(1111))
24 //*

```

1. Execution of DB2 Cloning Tool main program.
2. DB2 Cloning Tool LOAD library must be authorized.
3. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the DB2 Cloning Tool programs.
4. DD for CKZPRINT output.
5. Journal data set used by DB2 Cloning Tool commands. This VSAM data set is used to pass information between DB2 Cloning Tool steps and logs information derived from the step executions. The journal file is allocated and cataloged in the COPY step, and used as input and output by the RENAME

step, and input by the DB2UPDATE step. The specified data set name must match the data set allocated in the COPY step. If multiple DB2 Cloning Tool processes are created for different applications, each must use a unique journal data set. In the sample JCL, the control statement JOURNAL-DDN(JOURNAL) specifies that a DD with the name JOURNAL be used, rather than a data set name.

6. DD names 'BSDS01' and 'BSDS02' pointing to names of primary target member's BSDSs.
7. DD name 'DBD01' pointing to name of target 'DSNDBC.DSNDB01.DBD01' data set.
8. CKZIN DD control statement input.
9. DB2 Cloning Tool DB2UPDATE command. This command may be specified in the same execution with COPY and RENAME if desired. When multiple commands are specified in a single execution, if any command completes with a return code GE 8, the remaining commands are flushed without executing.
10. DB2-HLQS parameter specifying the source and target high level qualifier pairs of SRCDB21,TGTDB21 and SRCDB22,TGTDB22.
11. DB2-GROUP parameter specifying the source and target DB2 group names.
12. DB2-MEMBERS parameter specifying the source and target DB2 member names of DB1S, DB1T and DB2S, DB2T.

DB2UPDATE step JCL example – For CKZDUPD2

For completeness and to illustrate where JCL parameters and DB2 Cloning Tool control statements must match, the following JCL includes sample DB2 Cloning Tool control statements.

The DB2UPDATE step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

    //?????? JOB , 'DB2UPDATE2', CLASS=A, MSGCLASS=X
1 //S1      EXEC PGM=CKZ00010, REGION=6M
2 //STEPLIB DD DSN=HLQ?.SCKZLOAD, DISP=SHR
3 //CKZINI  DD DSN=HLQ?.SCKZPARM(CKZINI), DISP=SHR
4 //CKZPRINT DD SYSOUT=*
  //SYSUDUMP DD SYSOUT=*
5 //JOURNAL DD DSN=CKZ.JRNL, DISP=OLD
  //*****
  /* THE DB2 target2 MUST BE CHANGED TO MATCH THE HIGH LEVEL QUALIFIER
  /* FOR THE TARGET SYSTEM
  //*****
6 //BSDS01  DD DISP=OLD, DSN=target2.BSDS01
6 //BSDS02  DD DISP=OLD, DSN=target2.BSDS02
7 //CKZIN   DD *
8   DB2UPDATE
5     JOURNAL-DDN(JOURNAL)
9     DB2-HLQS(SRCDB21, TGTDB21, SRCDB22, TGTDB22)
10    DB2-GROUP(DBGS, DBGT)
11    DB2-MEMBERS(DB1S, DB1T,
                DB2S, DB2T)
      DDF(LUNAME(DB2TLU) PORT(22222)
12    BSDSONLY
  //*
```

1. Execution of DB2 Cloning Tool main program.
2. DB2 Cloning Tool LOAD library must be authorized.

3. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the DB2 Cloning Tool programs.
4. DD for CKZPRINT output.
5. Journal data set used by DB2 Cloning Tool commands. This VSAM data set is used to pass information between DB2 Cloning Tool steps and logs information derived from the step executions. The journal file is allocated and cataloged in the COPY step, and used as input and output by the RENAME step, and input by the DB2UPDATE step. The specified data set name must match the data set allocated in the COPY step. If multiple DB2 Cloning Tool processes are created for different applications, each must use a unique journal data set. In the sample JCL, the control statement JOURNAL-DDN(JOURNAL) specifies that a DD with the name JOURNAL be used, rather than a data set name.
6. DD names 'BSDS01' and 'BSDS02' pointing to names of secondary target member's BSDSs.
7. CKZIN DD control statement input.
8. DB2 Cloning Tool DB2UPDATE command. This command may be specified in the same execution with COPY and RENAME if desired. When multiple commands are specified in a single execution, if any command completes with a return code GE 8, the remaining commands are flushed without executing.
9. DB2-HLQS parameter specifying the source and target high level qualifier pairs of SRCDB21,TGTDB21 and SRCDB22,TGTDB22.
10. DB2-GROUP parameter specifying the source and target DB2 group names.
11. DB2-MEMBERS parameter specifying the source and target DB2 member names of DB1S, DB1T and DB2S, DB2T.
12. BSDSONLY parameter indicating only this member's BSDS data sets should be updated.

DB2UTILXCLEAN

This command is optional and is only used for the cloning of a DB2 system when you want to remove utility information from the target DB2 subsystem.

DB2UTILXCLEAN removes all entries from SYSUTILX. This command should be specified when utilities might be running or registered in SYSUTILX when the source DB2 subsystem is cloned. If SYSUTILX is not cleaned out, the source DB2 subsystem can become corrupted when the target DB2 subsystem cleans up the entries in SYSUTILX with a **-TERM UTIL** command. The DB2UTILXCLEAN command cleans out the target SYSUTILX and its indexes.

For more information on this command, see the following topics:

- “DB2 offline cloning” on page 95
- “DB2 online cloning” on page 104

If utilities were running or registered in SYSUTILX when the source DB2 subsystem was cloned, target objects might be in UT status after the cloning. The objects in UT status can be identified and **-STA DB(xxx) SP(yyy) ACCESS(FORCE)** commands can be issued against them to remove the UT status; however, this might cause the objects to be in an inconsistent state.

The CKZDUTST member of SCKZJCL can be used to run a REXX exec that identifies the table and index spaces in UT status and generates **-STA DB(yyy) SP(yyy) ACCESS(FORCE)** commands for the objects. These commands can be run to remove the UT status.

Note: DB2 Cloning Tool commands DB2ALTERBSDS, DB2LGRNXCLEAN, DB2RBLDBSDS, DB2SQL, DB2UPDATE, DB2UTILXCLEAN, and DB2XCFCLEAN must use the same value for DB2-NAME to correctly process for this DB2 subsystem or data-sharing group.

DB2UTILXCLEAN command syntax

DB2UTILXCLEAN

Required keywords:

{ JOURNAL-DSN(*data set name*) | JOURNAL-DDN(*ddname*) }

Optional keywords:

DB2-NAME(*name*)
SIMULATE

DB2UTILXCLEAN command and keyword definitions

Required keywords are described first, followed by optional keywords.

DB2UTILXCLEAN

Optional command to clean out DB2 SYSUTILX as part of the cloning.

- Required: No
- Restrictions: Used only for the cloning of a DB2 system when you want to remove utility information from the target DB2 subsystem.

JOURNAL-DSN (*data set name*)

or JOURNAL-DDN (*ddname*)

This parameter supplies either the data set name of the DB2 Cloning Tool journal file or the DD name of the DD statement in the JCL that points to the DB2 Cloning Tool journal file.

If multiple DB2 Cloning Tool setups are used for different volume groups, DO NOT use the same journal data set. Each DB2 Cloning Tool "application" needs a different journal data set.

The journal is used to pass information between DB2 Cloning Tool steps (for example, from the COPY step to the RENAME step). Therefore, as noted in the JCL comments, it must be cataloged in the COPY step when first created, and referenced as OLD in subsequent steps.

Do not delete the data set in the last step, in case restarts or reruns need journal information. It is preferable to delete and replace the data set only at the fresh start of a COPY.

- Default: None
- Required: Yes
- Restrictions: None

DB2-NAME(*name*)

Specifies a name that is the same name as the one used for the prior DB2UPDATE for this DB2 subsystem or data-sharing group.

Note: DB2 Cloning Tool commands DB2ALTERBSDS, DB2LGRNXCLEAN, DB2RBLDBSDS, DB2SQL, DB2UPDATE, DB2UTILXCLEAN, and

DB2XCFCLEAN must use the same value for DB2-NAME to correctly process for this DB2 subsystem or data-sharing group.

DB2-NAME is intended to be used when more than one DB2 subsystem or data-sharing group is being cloned from the same DB2 Cloning Tool COPY and RENAME. *Name* can be 1 - 4 alpha-numeric-national characters.

- Default: None
- Required: No
- Restrictions: None

SIMULATE

Specifies that the actions to clean out DB2 SYSUTILX are shown, but no modifications are made to SYSUTILX.

- Default: None
- Required: No
- Restrictions: If SIMULATE is not specified, the previous DB2UPDATE must not have been a SIMULATE.
- Short form: SIM

DB2UTILXCLEAN step JCL example

This topic contains an example of DB2UTILXCLEAN step JCL. Sample JCL can be found in the installation library SCKZJCL in member CKZDUTCL.

For completeness and to illustrate where JCL parameters and DB2 Cloning Tool control statements must match, the following JCL includes sample DB2 Cloning Tool control statements.

The DB2UTILXCLEAN step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to the notes that contain further information, following the sample JCL.

```
1 //??????? JOB , 'DB2UTILXCLEAN', CLASS=A, MSGCLASS=X
2 //S1 EXEC PGM=CKZ00010, REGION=8M
3 //STEPLIB DD DSN=HLQ?.SCKZLOAD, DISP=SHR
4 // DD DSN=DSNxxx.SDSNLOAD, DISP=SHR
5 //CKZINI DD DSN=HLQ?.SCKZPARM(CKZINI), DISP=SHR
6 //CKZPRINT DD SYSOUT=*
7 //SYSUDUMP DD SYSOUT=*
8 //JOURNAL DD DSN=CKZ.JRNL, DISP=OLD
9 //CKZIN DD *
10 DB2UTILXCLEAN -
11 JOURNAL-DDN(JOURNAL)
12 /**
```

1. Execution of DB2 Cloning Tool main program.
2. DB2 Cloning Tool LOAD library must be authorized.
3. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the DB2 Cloning Tool programs.
4. DD for CKZPRINT output.
5. Journal data set that is used by DB2 Cloning Tool commands. This VSAM data set is used to pass information between DB2 Cloning Tool steps and logs information that is derived from the step executions. The journal file is allocated and cataloged in the COPY step, and used as input and output by the RENAME step. The specified data set name must match the data set allocated in the COPY step. If multiple DB2 Cloning Tool setups are used for different

groups of volumes, ensure that you specify the journal data sets for each setup. In the sample JCL, the control statement JOURNAL-DDN(JOURNAL) specifies that a DD with the name JOURNAL is being used, instead of a data set name.

Step JCL example to identify objects with UT status

This topic contains an example of the step JCL for identifying objects with UT status. Sample JCL can be found in the installation library SCKZJCL in member CKZDUTST.

For completeness and to illustrate where JCL parameters and DB2 Cloning Tool control statements must match, the following JCL includes sample DB2 Cloning Tool control statements.

The step JCL for identifying objects with UT stats is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to the notes that contain further information, following the sample JCL.

```

|      //??????? JOB , 'CKZ CKZDUTST' ,CLASS=A,MSGCLASS=X
|      //S1      EXEC PGM=IDCAMS
|      //SYSPRINT DD  SYSOUT=*
|      //SYSIN    DD  *
|              DEL HLQ?.WRK.UTSTCMDS
|              SET MAXCC=0
|      /*
| 1 //S2      EXEC PGM=IKJEFT01,REGION=6M,PARM='CKZDUTST DB2T'
| 2 //STEPLIB DD  DSN=DSNxxx?.SDSNLOAD,DISP=SHR
| 3 //SYSEXEC DD  DISP=SHR,DSN=HLQ?.SCKZPARM
|      //SYSTSIN DD  DUMMY
|      //SYSTSPRT DD  SYSOUT=*
| 4 //CMDOUT  DD  DSN=HLQ?.WRK.UTSTCMDS,
|      //          UNIT=SYSDA,DISP=(,CATLG),
|      //          DSORG=PS,LRECL=80,RECFM=FB,BLKSIZE=0,
|      //          SPACE=(TRK,(1,1))
|      /*
|      /* IF IT IS DESIRED TO EXECUTE THE GENERATED COMMANDS THE
|      /* FOLLOWING STEP CAN BE UNCOMMENTED AND USED.
|      /*
|      /*IF1 IF (S2.RC LE 4) THEN
|      /*
|      /* EXECUTE THE GENERATED START DB FORCE COMMANDS
|      /*
| 5 /*S3      EXEC PGM=IKJEFT01,REGION=6M
|      /*STEPLIB DD  DISP=SHR,DSN=DSNXXX?.SDSNLOAD
|      /*SYSTSPRT DD  SYSOUT=*
|      /*SYSTSIN DD  DSP=SHR,DSN=HLQ?.WRK.UTSTCMDS
|      /*
|      /*IF1 ENDIF
|      /*

```

1. Execution of CKZDUTST REXX exec. DB2T is the target DB2 system to use.
2. DB2 LOAD library.
3. DD for SCKZPARM data set. The HLQ?.SCKZPARM library contains the CKZDUTST REXX exec.
4. DD for CMDOUT output. This DD will contain the generated **-STA DB(xxxxxxxx) SP(yyyyyyy) ACCESS(FORCE)** commands.
5. This step runs the generated commands. It is commented out to allow for manual checking of the generated commands before running them.

DB2XCFCLEAN

This command is optional and only used for the cloning of a DB2 data sharing system.

DB2XCFCLEAN deallocates the target DB2 data-sharing group XCF structures and removes the DB2 XCF group members.

The target DB2 group name comes from the prior DB2UPDATE command and is passed via the journal data set.

For more information about this command, see the following topics:

- “DB2 online cloning with removal of data sharing members procedure” on page 111
- “DB2 online cloning with target becoming non-data sharing procedure” on page 117

Note: DB2 Cloning Tool commands DB2ALTERBSDS, DB2LGRNXCLEAN, DB2RBLDBSDS, DB2SQL, DB2UPDATE, DB2UTILXCLEAN, and DB2XCFCLEAN must use the same value for DB2-NAME to correctly process for this DB2 subsystem or data-sharing group.

DB2XCFCLEAN command syntax

DB2XCFCLEAN

Required keywords:

{ JOURNAL-DSN(*data set name*) | JOURNAL-DDN(*ddname*) }

Optional keywords:

DB2-NAME(*name*)
SIMULATE

DB2XCFCLEAN command and keyword definitions

Required keywords are described first, followed by optional keywords.

DB2XCFCLEAN

Optional command to clean out DB2 XCF structures and group members.

- Required: No
- Restrictions:Used only for the cloning of a DB2 data sharing system.

JOURNAL-DSN (*data set name*)

or JOURNAL-DDN (*ddname*)

This parameter supplies either a data set name of the DB2 Cloning Tool journal file or a ddname assumed via the JCL to point at a journal data set. The journal data set for the RENAME step must be the same data set specified for the COPY step.

If multiple DB2 Cloning Tool setups are used for different volume groups, DO NOT use the same journal data set. Each DB2 Cloning Tool "application" needs a different journal data set.

The journal is used to pass information between DB2 Cloning Tool steps (for example, from the COPY step to the RENAME step). Therefore, as noted in the JCL comments, it must be cataloged in the COPY step when first created, and referenced as OLD in subsequent steps.

In case restarts or reruns need journal information, do not delete the data set in the last step. It is preferable to delete and replace the data set only at the fresh start of a COPY.

- Default: None
- Required: Yes
- Restrictions: None

DB2-NAME(*name*)

Specifies a name that is the same name as the one used for the prior DB2UPDATE for this DB2 subsystem or data-sharing group.

Note: DB2 Cloning Tool commands DB2ALTERBSDS, DB2LGRNXCLEAN, DB2RBLDBSDS, DB2SQL, DB2UPDATE, DB2UTILXCLEAN, and DB2XCFCLEAN must use the same value for DB2-NAME to correctly process for this DB2 subsystem or data-sharing group.

DB2-NAME is intended to be used when more than one DB2 subsystem or data-sharing group is being cloned from the same DB2 Cloning Tool COPY and RENAME. Name can be 1 - 4 alpha-numeric-national characters.

- Default: None
- Required: No
- Restrictions: None

SIMULATE

Specifies that which DB2 XCF structures and group members need cleaning is printed, but no modifications are made.

- Default: None
- Required: No
- Restrictions: If SIMULATE is not specified, the previous DB2UPDATE must not have been a SIMULATE.
- Short form: SIM

DB2XCFCLEAN step JCL example

This topic contains an example of DB2XCFCLEAN step JCL. Sample JCL can be found in the installation library SCKZJCL in member CKZDXCFC.

For completeness and to illustrate where JCL parameters and DB2 Cloning Tool control statements must match, the following JCL includes sample DB2 Cloning Tool control statements.

The DB2XCFCLEAN step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```
1 //??????? JOB , 'DB2XCFCLEAN', CLASS=A, MSGCLASS=X
2 //S1 EXEC PGM=CKZ00010, REGION=8M
3 //STEPLIB DD DSN=HLQ?.SCKZLOAD, DISP=SHR
4 //CKZINI DD DSN=HLQ?.SCKZPARM(CKZINI), DISP=SHR
5 //CKZPRINT DD SYSOUT=*
   //SYSUDUMP DD SYSOUT=*
6 //JOURNAL DD DSN=CKZ.JRNL, DISP=OLD
   //CKZIN DD *
   DB2XCFCLEAN
   JOURNAL-DDN(JOURNAL)
7 //*
```

1. Execution of DB2 Cloning Tool main program.

2. DB2 Cloning Tool LOAD library must be authorized.
3. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the DB2 Cloning Tool programs.
4. DD for CKZPRINT output.
5. Journal data set used by DB2 Cloning Tool commands. This VSAM data set is used to pass information between DB2 Cloning Tool steps and logs information derived from the step executions. The journal file is allocated and cataloged in the COPY step, and used as input and output by the RENAME step, and input by the DB2XCFCLEAN step. The specified data set name must match the data set allocated in the COPY step. If multiple DB2 Cloning Tool processes are created for different applications, each must use a unique journal data set. In the sample JCL, the control statement JOURNAL-DDN(JOURNAL) specifies that a DD with the name JOURNAL be used, rather than a data set name.

FINDUCATS

This command is not required. FINDUCATS will identify catalogs involved with source volume data sets. It does not negate the need to specify source and target catalog pairs in the COPY command.

Run FINDUCATS, at least initially, to determine involved ICF user catalogs, and then whenever you may wish to verify that the ICF user catalogs involved with source volume data sets are as specified in the COPY command.

FINDUCATS command syntax

FINDUCATS

Required keywords:

```
FROM-STORAGEGROUP(storgrp1 [ , storgrp2 ] [ , ... ] ) |
FROM-VOLSER( volser1 | volmask [ , volser2 | volmask ] [ , ... ] )
```

Optional keywords:

```
EXCLUDE-FROM-VOLSER( volser1 [ volser2 ] [ volsernm ] )
```

FINDUCATS command and keyword definitions

Required keywords are described first, followed by optional keywords.

FINDUCATS

Optional command to locate catalogs involved with source volume data sets.

- Required: No
- Restrictions: None

FROM-STORAGEGROUP (*storgrp1* [, *storgrp2*] [, ...])
or FROMSTORAGEGROUP

Specifies the input volumes to be scanned for involved ICF user catalogs. All volumes defined to the specified storage group(s) will be included in the scan. A warning will be generated for any volumes not found but the scan will continue

- Default: None
- Required: Yes
- Restrictions: FROM-STORAGEGROUP is mutually exclusive with FROM-VOLSER.

- Short form: FRS

FROM-VOLSER (*volser1* | *volmask* [, *volser2* | *volmask*] [, ...])
or FROMVOLSER

Specifies the input volumes to be scanned for involved ICF user catalogs.

- Default: None
- Required: Yes
- Restrictions: FROM-VOLSER is mutually exclusive with FROM-STORAGEGROUP.
- Short form: FRV

EXCLUDE-FROM-VOLSER (*volser1* [*volser2*] [*volsernmn*])
or EXCLUDEFROMVOLSER

Specifies volumes, or a volume mask for volumes to be excluded from those specified for the FROM-VOLSER parameter or the FROM-STORAGEGROUP parameter.

- Default: None
- Required: No
- Restrictions: None.
- Short form: EXCFRV

FINDUCATS step JCL example

This topic contains an example of FINDUCATS step JCL. Sample JCL can be found in the installation library SCKZJCL in member CKZFUCAT.

For completeness and to illustrate where JCL parameters and DB2 Cloning Tool control statements must match, the following JCL includes sample DB2 Cloning Tool control statements.

The FINDUCATS step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

1 //S1      EXEC PGM=CKZ00010,REGION=8M
2 //STEPLIB DD DSN=HLQ?.SCKZLOAD,DISP=SHR
3 //CKZINI  DD DSN=HLQ?.SCKZPARM(CKZINI),DISP=SHR
4 //CKZPRINT DD SYSOUT=*
  //SYSUDUMP DD SYSOUT=*
  //CKZIN   DD *
      FINDUCATS
      FROM-VOLSER(ABC*)
      EXCLUDE-FROM-VOLSER(ABCTST)
  //*
```

1. Execution of DB2 Cloning Tool main program.
2. DB2 Cloning Tool SCKZLOAD library must be authorized.
3. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the DB2 Cloning Tool programs.
4. DD for DB2 Cloning Tool output. Two reports are generated:
 - Report 1 – lists each volume serial number, the ICF user catalog names(s) involved, and the aliases or high level qualifiers on the volume.
 - Report 2 – summarizes the involved ICF user catalog names. Use the catalogs found to determine the source catalogs required in the USERCATALOGS parameter of the COPY command. For each source ICF

user catalog, you will need to supply a corresponding target catalog name to satisfy the USERCATALOGS requirement for source and target catalog pairs.

The alias information from report 1 will be useful in determining the target catalog aliases that must be created.

Note: FINDUCATS invokes DCOLLECT to identify ALIAS names of the source volume data sets in order to identify the correct source ICF user catalogs. DCOLLECT controls access to the DCOLLECT function, by issuing a security (RACF) check for a facility class profile of STGADMIN.IDC.DCOLLECT. If this profile exists, then Read authority is necessary. The command will not be successful if the user is not authorized (DB2 Cloning Tool does not make the call).

JRNLUPTGRADE

This command is not required. JRNLUPTGRADE is intended for situations where a current release of the DB2 Cloning Tool RENAME, DB2UPDATE, or BCSCLEAN command needs to use a journal data set created by an older release of DB2 Cloning Tool.

DB2 Cloning Tool commands that use the journal (except for the COPY and JRNLUPTGRADE commands) can only process a journal that was created by the same release of DB2 Cloning Tool.

Sometimes a journal created by a prior DB2 Cloning Tool release needs to be used by the current release. The JRNLUPTGRADE command can be used to upgrade a journal that was created by a prior release to the current release.

There are three scenarios for using JRNLUPTGRADE:

- Scenario 1: DB2 Cloning Tool COPY was run using a prior release and you want to run the current release of the DB2 Cloning Tool RENAME command.
- Scenario 2: DB2 Cloning Tool COPY and RENAME were run using a prior release and you want to run the current release of the DB2 Cloning Tool DB2UPDATE command and other DB2 conditioning commands.
- Scenario 3: The journal was created by a prior release and you want to run the current release of the DB2 Cloning Tool BCSCLEAN command.

It is not valid to run JRNLUPTGRADE between multiple runs of the RENAME command.

After running the JRNLUPTGRADE command, you might want to rename the old and new journal data sets, so that the new journal data set has the same name as the prior old journal. This way, you will not have to change the JCL in existing DB2 Cloning Tool cloning jobs.

Scenario 1: For scenario 1, the steps are:

1. Using the previous release of DB2 Cloning Tool: Run the DB2 Cloning Tool COPY command.
2. Back up the journal, the UCATBKUP data sets, and the target volumes.
3. At a later date or at a different site: Restore the journal, the UCATBKUP data sets, and the target volumes.
4. Using the current release of DB2 Cloning Tool: Run the DB2 Cloning Tool JRNLUPTGRADE command.

5. Using the current release of DB2 Cloning Tool: Run the DB2 Cloning Tool RENAME command.
6. For DB2, using the current release of DB2 Cloning Tool: Run the DB2 Cloning Tool DB2 conditioning commands.

Scenario 2: For scenario 2, the steps are:

1. Using the previous release of DB2 Cloning Tool: Run the DB2 Cloning Tool COPY command.
2. Using the previous release of DB2 Cloning Tool: Run the DB2 Cloning Tool RENAME command.
3. Using the current release of DB2 Cloning Tool: Run the DB2 Cloning Tool JRNLUPGRADE command.
4. Using the current release of DB2 Cloning Tool: Run the DB2 Cloning Tool DB2UPDATE command. Using the current release of DB2 Cloning Tool: Run the other DB2 Cloning Tool DB2 conditioning commands.

Scenario 3 can be used when a new release of DB2 Cloning Tool is being installed and repetitive clones are being run. For the repetitive clones, run BCSCLEAN before COPY for the next iteration. If the journal records used by BCSCLEAN have changed in the new release, the new release BCSCLEAN will fail. For this scenario, run JRLNUPGRADE before BCSCLEAN.

JRNLUPGRADE command syntax

JRNLUPGRADE

Required keywords:

```
{ NEW-JOURNAL-DDN( ddname ) | NEW-JOURNAL-DSN( data set name ) }
{ OLD-JOURNAL-DDN( ddname ) | OLD-JOURNAL-DSN( data set name ) }
```

JRNLUPGRADE command and keyword definitions

Required keywords are described first, followed by optional keywords.

JRNLUPGRADE

Optional command to upgrade a journal that was created with a prior release of DB2 Cloning Tool.

- Required: No
- Restrictions: None

NEW-JOURNAL-DDN (*ddname*)

or NEW-JOURNAL-DSN (*data set name*)

This parameter supplies either the data set name of the new DB2 Cloning Tool journal file, or the DD name of the DD statement in the JCL that points to the new DB2 Cloning Tool journal file.

- Default: None
- Required: Yes
- Restrictions: None.

OLD-JOURNAL-DDN (*ddname*)

or OLD-JOURNAL-DSN (*data set name*)

This parameter supplies either the data set name of the old DB2 Cloning Tool journal file, or the DD name of the DD statement in the JCL that points to the old DB2 Cloning Tool journal file.

- Default: None

- Required: Yes
- Restrictions: None.

JRNLUPGRADE step JCL example

This topic contains an example of JRNLUPGRADE step JCL. Sample JCL can be found in the installation library SCKZJCL in member CKZJRNU.

For completeness and to illustrate where JCL parameters and DB2 Cloning Tool control statements must match, the following JCL includes sample DB2 Cloning Tool control statements.

The JRNLUPGRADE step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

1 //??????? JOB ,CKZ JRNLUPGRADE',CLASS=A,MSGCLASS=X
  //S0      EXEC PGM=IDCAMS
  //SYSPRINT DD  SYSOUT=*
  //SYSIN   DD  *
2   DEL CKZ.UPGRADED.JRNL
   SET MAXCC=0
3 //S1      EXEC PGM=CKZ00010,REGION=6M
4 //STEPLIB DD  DSN=HLQ?.SCKZLOAD,DISP=SHR
5 //CKZINI  DD  DSN=HLQ?.SCKZ.PARM(CKZINI),DISP=SHR
  //CKZPRINT DD  SYSOUT=*
  //SYSUDUMP DD  SYSOUT=*
6 //OLDJRNL DD  DSN=CKZ.JRNL,DISP=SHR
7 //NEWJRNL DD  DSN=CKZ.UPGRADED.JRNL,
  //          RECOG=KS,KEYLEN=64,KEYOFF=0,
  //          DISP=(,CATLG),UNIT=SYSALLDA,
  //          LRECL=600,SPACE=(CYL,(10,10))
  //CKZIN   DD  *
   JRNLUPGRADE
6   OLD-JOURNAL-DDN(OLDJRNL)
7   NEW-JOURNAL-DDN(NEWJRNL)
  //*
```

1. Deletes any previously existing new journal data set, in anticipation of JRNLUPGRADE allocating a new upgraded journal for subsequent use by DB2 Cloning Tool. Because this data set is used to pass information from one DB2 Cloning Tool step to another, do not delete the new upgraded journal data set in any steps except the JRNLUPGRADE step.
2. Execution of DB2 Cloning Tool main program.
3. DB2 Cloning Tool SCKZLOAD library must be authorized.
4. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the DB2 Cloning Tool programs.
5. DD for DB2 Cloning Tool output.
6. The old journal data set that is to be upgraded. In the sample JCL, the control statement OLD-JOURNAL-DDN(OLDJRNL) specifies that a DD with the name OLDJRNL is being used rather than a data set name.
7. The new upgraded journal data set that is created by JRNLUPGRADE and will be used by subsequent DB2 Cloning Tool commands. In the sample JCL, the control statement NEW-JOURNAL-DDN(NEWJRNL) specifies that a DD with the name NEWJRNL is being used rather than a data set name.

ONLINECLIP

This command is not required. ONLINECLIP is intended for situations where the process used to copy the volumes (DB2 Cloning Tool COPY was not used) does not change the internal label of the target volumes back to the target volume serial; that is, the internal label reflects the source volume serial.

For DB2 Cloning Tool RENAME to function properly, the internal label needs to be corrected to contain the target volume serial.

ONLINECLIP will do this function. It will expect to find the target volumes online. It will read the internal label of the target device. If it has the corresponding source volume serial, it will be changed to the target volume serial.

For example, if you use the TSO FCESTABL command with the ONLINETGT=YES option, the target volume starts out with a target volume label. During the TSO FCESTABL, the target volume will remain online, but the volume label from the source volume will be copied to the target volume. The UCB however, still retains the target volume name. DB2 Cloning Tool RENAME requires the target volume to retain the target volume label. The DB2 Cloning Tool ONLINECLIP command reads the journal file to identify the original source and target volume pairs, scans the UCBs for the target volume names, and will change the target label so that it matches the original target volume serial number. This works because the system 'thinks' the target VOLSER is still online; it does not know the label at that device number was changed by FlashCopy. After the ONLINECLIP command has been executed, the target volumes are in the condition expected by DB2 Cloning Tool RENAME.

ONLINECLIP command syntax

ONLINECLIP

Required keywords:

{ JOURNAL-DSN(*data set name*) | JOURNAL-DDN(*ddname*) }

Optional keywords:

WRONG-VOLSER(RC(4 | 8))

ONLINECLIP command and keyword definitions

Required keywords are described first, followed by optional keywords.

ONLINECLIP

Optional command used to relabel the target volume(s) when the source volume label was copied but the UCB field still points to the target volume label. This can occur when TSO FCESTABL was used.

- Required: No
- Restrictions: None

JOURNAL-DSN (*data set name*)
or **JOURNAL-DDN** (*ddname*)

This parameter supplies either the data set name of the DB2 Cloning Tool journal file, or a DD name assumed via the JCL to point at a journal data set. If multiple DB2 Cloning Tool setups are used for different volume groups, DO NOT use the same journal data set. Each DB2 Cloning Tool 'application' needs a different journal data set.

The journal is used to pass information between DB2 Cloning Tool steps (e.g., from the COPY step to the RENAME step). Therefore, as noted in the JCL comments, it must be cataloged in the COPY step when first created, and referenced as OLD or SHR in subsequent steps.

- Default: None
- Required: Yes
- Restrictions: None.
- Short form(s): JDSN, or JDDN

WRONG-VOLSER(RC(4 | 8))

This option addresses the situation where the internal VOLSER of a target volume does not have the expected value. The ONLINECLIP command return code for this situation will be the specified value.

- Default: 4
- Required: No
- Restrictions: None.

ONLINECLIP step JCL example

| This topic contains an example of ONLINECLIP step JCL. Sample JCL can be
| found in the installation library SCKZJCL in member CKZOCLIP.

For completeness and to illustrate where JCL parameters and DB2 Cloning Tool control statements must match, the following JCL includes sample DB2 Cloning Tool control statements.

The ONLINECLIP step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

      //??????? JOB , 'ONLINECLIP', CLASS=A, MSGCLASS=X
1 //S1      EXEC PGM=CKZ00010, REGION=8M
2 //STEPLIB DD DSN=HLQ?.SCKZLOAD, DISP=SHR
3 //CKZINI  DD DSN=HLQ?.SCKZPARM(CKZINI), DISP=SHR
4 //CKZPRINT DD SYSOUT=*
      //SORTMSG DD SYSOUT=*
      //SYSUDUMP DD SYSOUT=*
5 //JOURNAL DD DSN=CKZ.JRNL, DISP=OLD
      //CKZIN  DD *
      ONLINECLIP
5      JOURNAL-DDN(JOURNAL)
      //*

```

1. Execution of DB2 Cloning Tool main program.
2. DB2 Cloning Tool LOAD library must be authorized.
3. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the DB2 Cloning Tool programs.
4. DD for CKZPRINT output.
5. Journal data set used by DB2 Cloning Tool commands. This VSAM data set passes information between DB2 Cloning Tool steps, and logs information derived from the step executions. The journal file is allocated and cataloged in the COPY step and used as input to the ONLINECLIP command. In the sample JCL, the control statement JOURNAL-DDN(JOURNAL) is specifying that a DD with the name JOURNAL is being used rather than a data set name.

RENAME

The *RENAME* command is required. The *RENAME* step renames and catalogs the data sets from the *COPY* step onto target volumes.

RENAME-MASKS are processed in order. The first hit of the source data set name is the one that is used for the target data set name.

Only the DB2 logs and BSDSs can be renamed beyond the *hlq*. All other DB2 files as shown in the following figure expect the following:

- DB2 directory – DB2 expects a specific naming standard:
*hlq.DSNDBx.DSNDB01.**
- DB2 catalog – DB2 expects a specific naming standard: *hlq.DSNDBx.DSNDB06.**
- DB2 databases – DB2 expects a specific naming standard –
vcat.DSNDBx.dbname.psname.y0001.Annn

RENAME command syntax

RENAME

Required keywords:

```
{ JOURNAL-DSN( data set name ) | JOURNAL-DDN( ddname ) }  
{ RENAME-MASKS( mask pairs ) | RENAME-MASKS-DDN( ddname ) }
```

Optional keywords:

```
DATACLAS( data class | SOURCE ) |  
DATACLAS-PAIRS(  
    Sourcedataclas1, Targetdataclas1,  
    Sourcedataclas2, Targetdataclas2, ...  
DEFAULT_IF_NO_MATCH,  
    Targetdataclasd | SOURCE )  
DRIVEACS  
EXCLUDE-SRCNAME( RC( 0 | NOTRENAMED-RC ) )  
EXCLUDE-SRCNAME-MASKS( masks ) | EXCLUDE-SRCNAME-MASKS-DDN( ddname )  
GDG-ALL-MIGRATED( SKIP ) | GDG-ALL-MIGRATED( RETAIN, RC( 0 | 4 ) )  
GDG-EMPTY( SKIP ) | GDG-EMPTY( RETAIN, RC( 0 | 4 ) )  
GDG-MIGRATED( ERROR ) | GDG-MIGRATED( RETAIN, RC( 0 | 4 ) )  
GDG-TAPE( ERROR ) | GDG-TAPE( RETAIN, RC( 0 | 4 ) )  
ISSUE-VCLOSE( NO ) | ISSUE-VCLOSE( YES | BEFORE | AFTER [ , LOCAL | SYSPLEX ] )  
MAX-TASKS( nnn | 1 )  
MGMTCLAS( mgmt class | SOURCE ) |  
MGMTCLAS-PAIRS(  
    Sourcemgmtclas1, Targetmgmtclas1,  
    Sourcemgmtclas2, Targetmgmtclas2, ...  
DEFAULT_IF_NO_MATCH,  
    Targetmgmtclasd | SOURCE )  
MISSINGUCAT( DELETE | KEEP [ , RC( 0 | 4 | 8 ) ] )  
NOTRENAMED( DELETE | KEEP [ , RC( 0 | 4 | 8 ) ] )  
ORPHANCATENTRY( DELETE | KEEP [ , RC( 0 | 4 | 8 ) ] )  
RECATALOG( Y | N )  
RENAME-AUDIT-LOG( N | SMF( nnn ) )  
RENAME-ERROR( ABORT ) | RENAME-ERROR( CONTINUE, RC( 0 | 4 | 8 ) )  
RENAME-LIST( Y | N )  
RERUN  
SAFE | SPEED  
SIMULATE  
STORCLAS( stor class | SOURCE ) |  
STORCLAS-PAIRS(  
    Sourcestorclas1, Targetstorclas1,  
    Sourcestorclas2, Targetstorclas2, ...  
DEFAULT_IF_NO_MATCH,
```

```

      Targetstorclass | SOURCE )
TEMPDSN( DELETE | KEEP [ , RC ( 0 | 4 | 8 ) ] )
UPDATE-IAM-ASSOCIATIONS( Y | N )
VALIDATE-SMS-CLASSES( Y | N )
VOLBKUP-DDN( ddname )

```

RENAME considerations

- The work data sets and output data sets created by the RENAME job step cannot reside on target volumes.
- RENAME-MASKS are processed in order. The first hit of the source data set name is the one that is used for the target data set name.
- Name lengths: Whether changing a qualifier to a longer new name qualifier, or using the + feature to add qualifiers, be careful that new names do not exceed 44 characters or 35 for GDG base names.
- Catalog aliases and new names: Catalog aliases must be set up to match names resulting from renaming.
- Rename collisions: Be sure rename masks cannot cause two or more old names to rename to the same new name.
- Should all data sets on all volumes be renamed? If all data sets on all volumes copied should be renamed, use the NOTRENAMED option with a return code of 8, assuming the application using the target volumes requires less than an 8 return code from RENAME.
- If non-VSAM data set aliases are used, ensure that if a rename mask matches a data set, the same mask, or other masks, will match all aliases defined for the data set.
- Only the DB2 logs and BSDSs can be renamed beyond the *hlq*. All other DB2 files expect the following:
 - DB2 directory – DB2 expects a specific naming standard:
*hlq.DSNDBx.DSNDB01.**
 - DB2 catalog – DB2 expects a specific naming standard:
*hlq.DSNDBx.DSNDB06.**
 - DB2 databases – DB2 expects a specific naming standard:
vcat.DSNDBx.dbname.psname.y0001.Annn
- An ICF catalog can be renamed, but it will not be usable as an ICF catalog.

Oldname syntax

Use the old name filter mask to select the data sets to apply the RENAME new name mask against. The allowable filter characters are shown in the following table:

Table 57. Filter characters allowed for old name filter masks

Character	Description
*	A single asterisk represents exactly one DSN qualifier of any value. Example: *. Or *. or .* . * combined with valid DSN characters or % means 0 to nn characters of any value.

Table 57. Filter characters allowed for old name filter masks (continued)

Character	Description
**	A double asterisk represents 0 to <i>nn</i> DSN qualifiers of any value. For example: **. or **. or **. ** cannot appear with any other characters within a qualifier. Three or more adjacent * are not allowed within a qualifier. ** can be used more than once in an 'old name' mask. Example: **.abcd.** or abc.**.defg.**
%	A percent sign represents one non-blank character.
!	An exclamation point represents one national character. @ # \$
<	A less-than sign represents one non-numeric character, national symbols included.
>	A greater-than sign represents one numeric character.

For example:

Filter = **.PAYROLL*.*%23*.DATA

would match DSN = TLQ050.PAYROLL.CYCLE23.DATA

For information about filters and ACS masks, refer to the topic "Filtering pattern masks" on page 23.

Newname syntax

Use the new name mask to rename the data sets selected by the old name filter mask. The allowable filter characters are shown in the following table:

Table 58. Filter characters allowed for new name filter masks

Character	Description
*	A single asterisk represents exactly one DSN qualifier. * may not be used for a partial qualifier in a 'new name' mask. Example: aaa.*bb.** would not be valid.
**	A double asterisk represents 0 to <i>nn</i> DSN qualifiers of any value. For example: **. or **. or **. ** cannot appear with any other characters within a qualifier. Three or more adjacent * are not allowed within a qualifier. ** can be used more than once in an 'old name' mask. Example: **.abcd.** or abc.**.defg.**
%	A percent sign represents one non-blank character.
!	An exclamation point represents one national character. @ # \$
+cccccc	A plus sign followed by 1 to 8 characters means 'insert this new qualifier'.
-	A minus sign means 'remove this qualifier from the new name'.

For example:

Original DSN/Mask: CKZI.LAB9.DEM01.SDS.ORIG

Rename To DSN/Mask: CKZI.+TEST.XX%%.-.**

New Name: CKZI.TEST.XXB9.SDS.ORIG

For information about filters and ACS masks, refer to the topic “Filtering pattern masks” on page 23.

RENAME command and keyword definitions

Required keywords are described first, followed by optional keywords.

RENAME

Renames and catalogs data sets on target volumes.

- Required: Yes
- Restrictions: None

JOURNAL-DSN (*data set name*) or JOURNAL-DDN (*ddname*)

This parameter supplies either a data set name of the DB2 Cloning Tool journal file or a DD name assumed via the JCL to point at a journal data set. *The journal data set for the RENAME step must be the same data set specified for the COPY step.*

If multiple DB2 Cloning Tool setups are used for different volume groups, DO NOT use the same journal data set. Each DB2 Cloning Tool ‘application’ needs a different journal data set.

The journal is used to pass information between DB2 Cloning Tool steps (e.g., from the COPY step to the RENAME step). Therefore, as noted in the JCL comments, it must be cataloged in the COPY step when first created and referenced as OLD in subsequent steps. In case restarts or reruns need journal information, do not delete the data set in the last step. It is preferable to delete and replace the data set only at the fresh start of a COPY.

- Default: None
- Required: Yes
- Restrictions: None

RENAME-MASKS (*mask pairs*)

RENAME-MASKS are specified in ‘oldname’ ‘newname’ pairs. RENAME-MASKS are processed in order. The first hit of the source data set name is the one that is used for the target data set name.

RENAME-MASKS can be specified so that the target data set names will be exactly the same as the source data set names.

All data sets can be renamed beyond the hlq except for:

- DB2 directory – DB2 expects a specific naming standard:
hlq.DSNDBx.DSNDB01.*
- DB2 catalog – DB2 expects a specific naming standard:
hlq.DSNDBx.DSNDB06.*
- DB2 databases – DB2 expects a specific naming standard:
vcat.DSNDBx.dbname.psname.y0001.Annn
- DB2 BSDSs, and DB2 logs can be renamed beyond the hlq.

For example:

RENAME-MASKS (DB2APROD.** DB2CPROD.** –
DB2BPROD.** DB2DPROD.**)

- Default: None
- Required: Yes
- Restrictions: See topic “RENAME considerations” on page 434. Mutually exclusive with RENAME-MASKS-DDN.

RENAME-MASKS-DDN (*ddname*)

This parameter specifies a DD name that points to a file containing rename mask pairs. The pairs are the same format as in the RENAME-MASKS keyword. RENAME-MASKS-DDN must have an LRECL of 80, and must not contain sequence numbers in columns 73 - 80.

- Default: None
- Required: Yes
- Restrictions: Mutually exclusive with RENAME-MASKS.
- Short form: RM-DDN

DATACLAS (*data class* | **SOURCE**)

or **DATACLAS-PAIRS** (*Sourcedataclas1, Targetdataclas1, Sourcedataclas2, Targetdataclas2, ... DEFAULT_IF_NO_MATCH, Targetdataclasd* | **SOURCE**)

DATACLAS specifies the SMS DATACLAS to be used for all renamed data sets on SMS managed volumes if the value is in quotes, or will be copied from the corresponding source volume data set if **SOURCE** is specified. The default is **SOURCE**.

DATACLAS-PAIRS specifies source/target pairs for dataclas. If a data set has the specified source dataclas, the target data set will be given the paired target dataclas.

DEFAULT_IF_NO_MATCH indicates a dataclas to be assigned to any target data set whose source dataclas was not matched by any other **DATACLAS-PAIR** entry.

SOURCE indicates such data sets should be assigned the dataclas used by the source data set. If you want to use an SMS dataclas, **SOURCE**, enter **SOURCE** in quotes.

- Default: For **DATACLAS**, the default is **SOURCE**.
- Required: No
- Restrictions: **DATACLAS** and **DATACLAS-PAIRS** are mutually exclusive with **DRIVEACS**.
- Short form(s): DC, or DCP.

DRIVEACS

Specifies that SMS class information for renamed data sets is to be derived by 'driving' ACS routines. Note that variables supplied to ACS are: DSN, STORCLAS, DATACLAS, MGMTCLAS, ACSENVIR (RENAME), XMODE (BATCH), JOB_ACCT, STEP_ACCT, USERID, GROUP, APPLIC, SYSNAME, SYSPLEX, JOBNAME, PGM, and NAME.

The use of the **DRIVEACS** parameter may significantly slow down the **RENAME** performance.

- Default: None
- Required: No
- Restrictions: **DRIVEACS** is mutually exclusive with **DATACLAS**, **DATACLAS-PAIRS**, **MGMTCLAS**, **MGMTCLAS-PAIRS**, **STORCLAS**, and **STORCLAS-PAIRS**.

Note: Variables supplied to ACS do not include data set attributes because DB2 Cloning Tool catalogs target data sets concurrently with changing target volumes – as opposed to cataloging data sets only after volumes have been modified. This methodology greatly improves the performance of RENAME, at the expense of the cataloging task not being aware of data set attributes that are normally supplied when driving ACS.

If SMS treatment of target volume data sets will be special, specific SMS classes can be supplied with the DATACLAS, MGMTCLAS, and STORCLAS keywords, or you can also set up a construct that triggers the DB2 Cloning Tool job name.

EXCLUDE-SRCNAME(RC(0 | NOTRENAMED-RC))

This parameter specifies the return code that will be given for data sets that match entries in the EXCLUDE-SRCNAME-MASKS keyword. 0 specifies that a return code of zero will be given for data sets that match entries in the EXCLUDE-SRCNAME-MASKS keyword. NOTRENAMED-RC specifies that the RC specified in the NOTRENAMED keyword will be given for data sets that match entries in the EXCLUDE-SRCNAME-MASKS keyword. Using RC(0) addresses the situation where there are known data sets on the volumes that will not be renamed and it is desired to use NOTRENAMED(RC(8)) to know if some not known data sets are on the volumes.

- Default: NOTRENAMED-RC
- Required: No
- Restrictions: Only used when EXCLUDE-SRCNAME-MASKS is specified.
- Short form: XS

EXCLUDE-SRCNAME-MASKS (*masks*)

This parameter supplies a list of source data set names or masks that will not be renamed (excluded from renaming).

Data sets that are not renamed due to this keyword will be treated as if there were no rename mask match. The disposition of these not-renamed data sets is controlled by the NOTRENAMED keyword and the return code generated is controlled by the EXCLUDE-SRCNAME keyword.

To exclude a GDG and its GDS entries, two masks should be used. One mask should be the GDG base name and the other mask should be the GDG base name plus G>>>>v>>.

For example: GDG.BASE.NAME GDG.BASE.NAME.G>>>>V>>

Using two masks of this form is necessary due the different ways the data set names are stored in the volume VTOC and in the ICF catalog.

- Default: None
- Required: No
- Restrictions: Mutually exclusive with EXCLUDE-SRCNAME-MASKS-DDN. See topic “RENAME considerations” on page 434.
- Short form: XSM

EXCLUDE-SRCNAME-MASKS-DDN(*ddname*)

This parameter specifies the DD name that points to a file containing EXCLUDE-SRCNAME-MASKS. The entries are the same format as in the

EXCLUDE-SRCNAME-MASKS keyword. EXCLUDE-SRCNAME-MASKS-DDN must have an LRECL of 80, and must not contain sequence numbers in columns 73 - 80.

- Default: None
- Required: No
- Restrictions: Mutually exclusive with EXCLUDE-SRCNAME-MASKS. See topic “RENAME considerations” on page 434.
- Short form: XSM-DDN

GDG-ALL-MIGRATED(SKIP)
or GDG-ALL-MIGRATED (RETAIN, RC(0 | 4))

This command supports DFSMSHsm, FDR, and CA-Disk. This option addresses the situation where a GDG matches a RENAME mask and all the source generations have been migrated.

The GDG entry can be skipped, or the GDS entries in the GDG base record may be RETAINED with a corresponding return code of 0 or 4. If not specified in the control statements, the defaults are obtained from the CKZINI member of SCKZPARM.

Note: If RETAIN is used, because the migrated generations do not exist under the new name, subsequent access to the generations will fail whether one is accessed specifically or via specification of the base name only. This option is provided to retain relativity.

Note: To avoid destroying the relativity of active generations, DB2 Cloning Tool does NOT allow removing selected generations. For data that is migrated and required on the target volumes, they must be recalled prior to the COPY.

- Default: The default is SKIP. For RETAIN, the default is RC(4)
- Required: No
- Restrictions: RC is mutually exclusive with SKIP.

GDG-EMPTY(SKIP)
or GDG-EMPTY (RETAIN, RC(0 | 4))

This option addresses an empty base GDG that matches a RENAME mask.

The GDG entry can be skipped, or the new base entry can be added to the target user catalog. If not specified in the control statements, the defaults are obtained from the CKZINI member of SCKZPARM.

- Default: The default is SKIP. For RETAIN, the default is RC(4)
- Required: No
- Restrictions: RC is mutually exclusive with SKIP.

GDG-MIGRATED(ERROR)
or GDG-MIGRATED (RETAIN, RC(0 | 4))

This option addresses the situation where a GDG matches a rename mask and at least one generation is indeed found on a volume, yet one or more generations are migrated. The migrated generation may be treated as an ERROR, or the GDS entry in the GDG base record may be RETAINED with a corresponding return code of 0 or 4.

If not specified in the control statements, the defaults are obtained from the CKZINI member of SCKZPARM.

Note: If RETAINED, because the migrated generation does not exist under the new name, subsequent access to the generation will fail whether it is accessed specifically or via specification of the base name only.

To avoid destroying the relativity of active generations, DB2 Cloning Tool does not allow removing selected generations. Retaining non-existent migrated generations may be suitable for situations such as overstated GDG limits where it is normal for older generations to be migrated and hopefully never accessed, or Log Files etc. where perhaps only the current generation is kept on primary and older migrated generations are kept as a safety factor.

- Default: The default is ERROR. For RETAIN, the default is RC(4)
- Required: No
- Restrictions: RC is mutually exclusive with ERROR.

GDG-TAPE(ERROR)

or **GDG-TAPE(RETAIN,RC(0 | 4))**

This option addresses the situation where a GDG matches a rename mask and at least one generation is indeed found on a volume, yet one or more generations are on tape. The tape generation may be treated as an ERROR, or the GDS entry in the GDG base record may be RETAINED with a corresponding return code of 0 or 4.

If not specified in the control statements, the defaults are obtained from the CKZINI member of SCKZPARM.

Note: If RETAIN is specified, accessing a target tape GDS will cause a S813 ABEND whether it is accessed specifically or via specification of the base name only.

To avoid destroying the relativity of active generations, CKZ does not allow removing selected generations.

Retaining non-existent tape generations may be suitable for situations such as overstated GDG limits where older generations may have been created on tape.

- Default: The default is ERROR. For RETAIN, the default is RC(4)
- Required: No
- Restrictions: RC is mutually exclusive with ERROR.

ISSUE-VCLOSE(NO)

or **ISSUE-VCLOSE (YES | BEFORE | AFTER [, LOCAL | SYSPLEX])**

Specifies whether a catalog modify command, F CATALOG,VCLOSE(targetvolser), will be issued as part of the volume RENAME processing. The Catalog Address Space (CAS), caches VVDS information. The modify command requests that the VVDS information cached for the target volume be refreshed.

NO specifies that the modify command will NOT be issued.

BEFORE specifies that the modify command will be issued only before the VVDS is updated.

AFTER specifies that the modify command will be issued only after the VVDS has been updated.

YES specifies that the modify command will be issued both before the VVDS is updated and after the VVDS has been updated.

If not specified in the control statements, the defaults are obtained from the CKZINI member of SCKZPARM.

LOCAL – the catalog modify command, F
CATALOG,VCLOSE(targetvolser), will be issued only on the system that
RENAME is running on.

SYSPLEX – the catalog modify command, F
CATALOG,VCLOSE(targetvolser), will be issued on the local system, and
the modify command will be routed to all the other systems in the sysplex,
via an MVS ROUTE *OTHER command, after the VVDS has been updated.

- Default: YES, LOCAL
- Required: No
- Restrictions: LOCAL and SYSPLEX are mutually exclusive with NO.

MAX-TASKS (*nnn* | 1)

Specifies the maximum subtasks to be used for volume processing in the
RENAME step. Increasing the number of subtasks can greatly reduce the
volume processing time. Provide a reasonably large region size and
increase the number of subtasks until a point of no gain is realized.
Because contention issues contributing to the point of no gain will vary by
installation, experiment with this parameter to determine the optimum
setting. Although the RENAME step can be rerun, because volume VTOCs,
VTOC indexes, and VVDSs are restored when RERUN is specified,
comparative times will be distorted. Hence, when experimenting, run the
process completely over from the COPY step. This exercise can also be a
good time to experiment with the COPY COPYCMDLIMIT option.

Another factor to consider is the cataloging time. The cataloging is
performed by a subtask of RENAME that runs in parallel with the volume
processing subtasks. If the number of data sets involved is high compared
to the number of volumes, because the cataloging subtask may take longer
than all volume processing subtasks, increasing the volume processing
MAX-TASKS may have no effect on the complete run time of RENAME.
The time the cataloging process ends, relative to the end time of the entire
step, can be found in the output.

The maximum value is 255.

- Default: 1
- Required: No
- Restrictions: None.

MGMTCLAS (*management class* | SOURCE)

**or MGMTCLAS-PAIRS (*Sourcemgmtclas1, Targetmgmtclas1, Sourcemgmtclas2,*
Targetmgmtclas2, ... DEFAULT_IF_NO_MATCH, Targetmgmtclasd | SOURCE)**

MGMTCLAS specifies the SMS MGMTCLAS to be used for all renamed
data sets on SMS managed volumes if the value is in quotes, or will be
copied from the corresponding source volume data set if SOURCE is
specified. SOURCE is the default. MGMTCLAS is mutually exclusive with
DRIVEACS.

MGMTCLAS-PAIRS specifies source/target pairs for mgmtclas.

If a data set has the specified source mgmtclas, the target data set will be
given the paired target mgmtclas.

DEFAULT_IF_NO_MATCH indicates a mgmtclas to be assigned to any
target data set whose source mgmtclas was not matched by any other
MGMTCLAS-PAIR entry.

SOURCE indicates such data sets should be assigned the mgmtclas used by the source data set. If you want to use an SMS mgmtclas, SOURCE, enter SOURCE in quotes.

- Default: The default for MGMTCLAS is SOURCE.
- Required: No
- Restrictions: MGMTCLAS and MGMTCLAS-PAIRS are mutually exclusive with DRIVEACS.
- Short form(s): MC, or MCP.

MISSINGUCAT (DELETE | KEEP [, RC (0 | 4 | 8)])

Specifies the disposition and return code to be generated for data sets found on a volume, where the data set name matches a rename mask, but the catalog back-pointer is not one of the 'source' catalogs specified in the corresponding COPY command. If not otherwise specified in the control statements, the default disposition and return code are obtained from this token in the CKZINI member of the SCKZPARM library.

Only data sets with VVDS records have a catalog back-pointer. A return code of 8 is suggested because MISSINGUCAT errors will likely be due to the omission of a catalog name in the COPY step.

Catalog back-pointers could be in error to start with. However, to use a return code of 8 and hence avoid continual review of any errors disclosed, it is preferable to diagnose volumes so that a MISSINGUCAT detection in fact means a user catalog was omitted in the COPY step.

- Default: KEEP,RC(4)
- Required: No
- Restrictions: None.

NOTRENAMED (DELETE | KEEP [, RC (0 | 4 | 8)])

Specifies the disposition of data sets that do not match a rename mask, and the return code to be generated if at least one occurrence is detected. If not otherwise specified in the control statements, the default disposition and return code are obtained from this token in the CKZINI member.

If you require that rename masks match all data sets on all volumes, and if they don't, you want the entire process to fail, specify a return code of 8 and the corresponding conditional use of the target volumes set for execution only if the return code is less than 8. In that the RENAME and or COPY step will be rerun, specify KEEP as the disposition.

If volumes contain data sets not needed by the application that will access target volumes, and you do not want to rename the non application data sets and want the entire process to fail if other data sets are on the volumes, specify a return code of 8 and also specify EXCLUDE-SRCNAME(RC(0)) and EXCLUDE-SRCNAME-MASKS with entries identifying all the expected non application data sets. If all the data sets on the volumes are renamed or match an entry in EXCLUDE-SRCNAME-MASKS the return code will be 0. If there is any data set on a volume that is not renamed and does not match an entry in EXCLUDE-SRCNAME-MASKS the return code will be 8 and the process will fail. Because the data sets that match an entry in EXCLUDE-SRCNAME-MASKS will not be renamed and they will also not be cataloged, especially for SMS-managed volumes, you should specify the DELETE option. This also frees up the space if allocations may occur on target volumes.

If volumes contain data sets not needed by the application that will access target volumes, and rename masks may not match all data sets, specify a

return code of 0 or 4. Because not-renamed data sets will also not be cataloged, especially for SMS-managed volumes, you should specify the DELETE option. This also frees up the space if allocations may occur on target volumes.

However, note that if not all data sets are renamed, and the return code is set for this to be acceptable (0 or 4), the list of not-renamed data sets will need to be reviewed for assurance that needed data sets are indeed renamed. Obviously, it is best if the application involved "owns" the volumes and hence all data sets should be renamed. In a compromise situation, where the volumes are used by other applications, if feasible, the advantage of renaming data sets that are not actually required, is that the NOTRENAMED return code can be set to 8 to avoid reviewing the list of not-renamed data sets each cycle.

This may be of little concern if a single, or very few rename masks will match all required data sets (i.e., you use effective naming conventions). If, for instance, all data sets belonging to the application will match the mask A1.**, it is probably a safe bet that ignoring data sets not matching the mask will not cause a problem.

DB2 Cloning Tool may be used to effectively copy only selected data sets, by not renaming all data sets. If renaming only selected data sets, with the NOTRENAMED option set to delete, the target volume will be left with only the desired data sets. Note that because the copies are by volume, track locations of target data sets will be the same as their source volume counterparts.

- Default: KEEP,RC(8)
- Required: No
- Restrictions: None.

ORPHANCATENTRY (DELETE | KEEP [, RC(0 | 4 | 8)])

Specifies the disposition and return code to be generated for data sets found in a catalog but one or more catalog volume cells are not in the list of volumes copied. If not otherwise specified in the control statements, the default disposition and return code are obtained from this token in the CKZINI member of the SCKZPARM library.

- Default: KEEP,RC(8)
- Required: No
- Restrictions: None.

RECATALOG(Y | N)

Specifies that DB2 Cloning Tool may replace an existing catalog entry without considering it an error. If RECATALOG is not specified, and a target catalog entry for a renamed data set is found to exist, the process fails. If not specified in the control statements, the defaults are obtained from the CKZINI member of SCKZPARM.

Because an incorrect rename 'to' mask could accidentally replace a catalog entry for a data set not involved with the process, the setup should be established such that RECATALOG is not required. This setup can be achieved by utilizing a target catalog used only for target data sets.

Because catalog entries for data sets used in each cycle of the process will be 'orphaned' by the target volumes being reused, at the beginning of the next cycle, delete and redefine the catalog, or use the DB2 Cloning Tool BCSCLEAN command to delete catalog entries from the previous cycle.

If a target catalog is populated with entries not involved with the copy process, use the BCSCLEAN command sometime between the time target volume usage ends and before the next copy process starts. By eliminating catalog entries from the previous cycle, omitting the RECATALOG option should not result in errors – unless rename 'to' masks are wrong, in which case the replacement of a catalog entry should indeed be prevented.

- Default: N
- Required: No
- Restrictions: None.

RENAME-AUDIT-LOG (N | SMF(*nnn*))

Specifies whether an audit log of the data sets being renamed is to be created by RENAME volume processing.

SMF(*nnn*) specifies that the audit log will be created and written to SMF with a record type of *nnn*. Valid values for *nnn* are 128 through 255 inclusive. SMF must be recording the specified record type. The layout of the records written can be found in member CKZRNSMF of the DB2 Cloning Tool JCL library.

- Default: N
- Required: No
- Restrictions: None.

RENAME-ERROR(ABORT)

or RENAME-ERROR(CONTINUE,RC(0 | 4 | 8))

This option specifies how processing proceeds when a RENAME error is encountered.

ABORT will terminate with an RC=8 after the first error to preserve integrity. ABORT is recommended.

CONTINUE will continue processing after most errors and the RENAME command will complete with the specified return code unless an error not handled by the CONTINUE logic is encountered.

Note: The use of CONTINUE can cause inconsistencies between the contents of the volumes and catalogs. Possible problems include:

- data sets could be cataloged but are not renamed on disk
- data sets could be renamed on disk but are not cataloged
- data sets that are not renamed on disk may not be deleted from disk
- GDG base and GDS entries will not exist in the catalog when there is a missing GDS
- a catalog entry may not point at the correct volume
- a catalog entry may be invalid
- leave uncataloged data sets on SMS managed volumes.

If this keyword is specified, DB2 Cloning Tool will not guarantee integrity and the given results will not be fixed by DB2 Cloning Tool.

If not specified in the control statements, the defaults are obtained from the CKZINI member of SCKZPARM.

- Default: ABORT. For CONTINUE, the default is RC(8)
- Required: No
- Restrictions: RC is mutually exclusive with ABORT.

RENAME-LIST(Y | N)

Specifies whether a list of the renamed data sets is to be produced by RENAME volume processing.

- Default: N if SIMULATE is not specified. Y if SIMULATE is specified.
- Required: No
- Restrictions: None.

RERUN

Specifies that the RENAME step is being run a second time using the same target volumes resulting from the COPY step. Rerun of the RENAME step only is not possible unless the first execution specified SAFE. See the SAFE | SPEED option for more information.

SAFE causes a backup during the RENAME step of the portions of volumes changed during the volume processing – VTOC, VTOCIX, and VVDS. This backup data is used on a rerun to restore any volumes changed to the state they were in following the COPY step.

Certain errors, such as incorrect rename masks, where the list of volumes copied and the catalogs backed up are correct, can be corrected by rerunning just the RENAME step.

Examine the problem to determine if just a rerun of the RENAME step will resolve the problem or if the COPY step must be rerun. Any errors resulting from a volume being omitted or a catalog being omitted from the COPY step will require a rerun of the COPY step.

The BCSRECS and VOLBKUP data sets must not be deleted by the JCL running the RENAME RERUN.

The JCL used for RENAME RERUN is different from the JCL used for RENAME SAFE. Sample JCL for RENAME RERUN can be found in the installation SCKZJCL library in member CKZRENRR.

The contents of the BCSRECS and VOLBKUP must be from the prior run of RENAME with either the SAFE or RERUN keywords. If the BCSRECS or VOLBKUP data sets are deleted, the COPY step will need to be run again.

- Default: None.
- Required: No
- Restrictions: RERUN is mutually exclusive with SAFE and SPEED.

SAFE | SPEED

SAFE allows a rerun of the RENAME command by backing up critical volume structures that are changed during the volume processing – the VTOC, VTOCIX, and VVDS. This backup data is used on a rerun to restore any volumes changed to the state they were in following the COPY step. This adds some slight execution time for RENAME to capture the portions of target volumes modified by RENAME.

Incorrect rename masks may be a reason for needing to rerun the RENAME step. If multiple and complicated masks are required, this option is recommended. Also affecting the renaming is whether the data set naming conventions used by the application are fairly static or subject to frequent change – inferring that rename masks need to be watched.

SAFE requires the VOLBKUP parameter. Note the JCL comments that warn about making sure the VOLBKUP data set is not deleted before a rerun of

RENAME. If the VOLBKUP data set is lost, the COPY step will need to be run again, provided that the opportunity for correct point-in-time images has not been lost.

If source volume access is not resumed until the entire process is complete (implying that the same point-in-time images can be re-copied), the time to rerun the COPY step may be insignificant compared to adding some overhead with the SAFE option for every cycle.

SPEED is the opposite of SAFE. The RERUN option for the RENAME step will be rejected if attempted. Correction of any errors will require the COPY and RENAME step to be run again.

- Default: SPEED.
- Required: No
- Restrictions: SAFE is mutually exclusive with SPEED and RERUN. SPEED is mutually exclusive with SAFE and RERUN.

SIMULATE

Specifies that only non-destructive RENAME activities are to be performed. RENAME SIMULATE must be used in conjunction with a COPY as the volumes and catalogs are specified only in the COPY step and are passed via the journal to the RENAME step. The COPY execution may be with or without SIMULATE.

Scenario 1 – a real COPY (without SIMULATE) followed by a RENAME SIMULATE will use only the target volumes.

Scenario 2 – a COPY SIMULATE followed by a RENAME SIMULATE will use only the source volumes because there are no target volumes yet.

If COPY DATAMOVER(PGM(NONE)) was used with SIMULATE, it can be followed by a RENAME SIMULATE.

SIMULATE will perform the masking comparisons to both catalog and volume records. Errors can be discovered such as not all data sets renamed, incomplete renaming of VSAM sphere associations and components, multivolume data sets not wholly contained in the volume list, GDG generations that stray outside of the volume list, catalog entries that match a mask but one or volumes were not 'copied', etc.

Because omitting a catalog(s) or volume(s) in the COPY step is not detected until the RENAME step, it is strongly advised that SIMULATE be used whenever significant changes are made to the involved application(s), especially if resumption of source volume access is triggered by completion of the COPY step – i.e., the opportunity for re-capturing point-in-time images has been lost.

- Default: None.
- Required: No.
- Restrictions: None.
- Short form: SIM

STORCLAS (*storage class* | **SOURCE**)

or **STORCLAS-PAIRS** (*Sourcestorclas1, Targetstorclas1, Sourcestorclas2, Targetstorclas2, ...* **DEFAULT_IF_NO_MATCH**, *Targetstorclasd* | **SOURCE**)

STORCLAS specifies the SMS STORCLAS to be used for all renamed data sets on SMS managed volumes if the value is in quotes, or will be copied from the corresponding source volume data set if SOURCE is specified.

STORCLAS-PAIRS specifies source/target pairs for storclas.

If a data set has the specified source storclas, the target data set will be given the paired target storclas.

DEFAULT_IF_NO_MATCH indicates a storclas to be assigned to any target data set whose source storclas was not matched by any other STORCLAS-PAIR entry.

SOURCE indicates such data sets should be assigned the storclas used by the source data set. If you want to use an SMS storclas, SOURCE, enter SOURCE in quotes.

- Default: The default for STORCLAS is SOURCE..
- Required: No.
- Restrictions: STORCLAS and STORCLAS-PAIRS are mutually exclusive with DRIVEACS.
- Short form(s): SC, or SCP

TEMPDSN (DELETE | KEEP [,RC(0 | 4 | 8)])

Specifies the disposition of temporary data sets and the return code to be generated if at least one occurrence is discovered. If not otherwise specified in the control statements, the default disposition and return code are obtained from this token in the CKZINI member of THE SCKZPARM LIBRARY.

- Default: DELETE,RC(4)
- Required: No.
- Restrictions: None.

UPDATE-IAM-ASSOCIATIONS (Y | N)

Specifies whether IAM data set associations are to be updated as part of RENAME processing. IAM must be active on the system for the updates to happen.

This option addresses the situation where there are IAM data sets that are being cloned that include AIXes and PATHs, and it is desired to update the associations to correspond with the new data set names. The association information for IAM data sets will be determined and updated by internally using IDCAMS LISTCAT and IDCAMS DEFINE RECATALOG commands.

When using RERUN, it is possible to get missing component errors when rename mask entries that cover IAM data sets have changed. This is due to the IAM association data not being in the ICF catalog, VTOC, VTOCIX, or VVDS, so DB2 Cloning Tool is unable to properly determine the changed IAM associations. When using SIMULATE, it is not possible for DB2 Cloning Tool to determine the IAM associations and verify that all the associated IAM data sets have been cloned.

- Default: N
- Required: No.
- Restrictions: None.
- Short form: UIA

VALIDATE-SMS-CLASSES (Y | N)

Specifies whether the SMS class names specified in the DATACLAS, DATACLAS-PAIRS, MGMTCLAS, MGMTCLAS-PAIRS, STORCLAS, and STORCLAS-PAIRS keywords will be validated as being defined to SMS (Y) or not (N).

This option addresses the situation where the target SMS class names are not defined on the system where RENAME is run.

- Default: Y
- Required: No.
- Restrictions: None.

VOLBKUP-DDN (*ddname*)

Specifies the DD name for the backup data set to be used for backing up target volume VTOCs, VTOCIXs, and VVDSs, to be used in the event of a rerun of the RENAME step. SAFE is required in conjunction with this parameter.

- Default: None
- Required: No.
- Restrictions: None.

RENAME step JCL example

This topic contains an example of RENAME step JCL for RENAME SAFE. Sample JCL can be found in the installation library SCKZJCL in member CKZREN.

The JCL used for RENAME RERUN is different from the JCL used for RENAME SAFE. Sample JCL for RENAME RERUN can be found in the installation library SCKZJCL in member CKZRENRR.

For completeness and to illustrate where JCL parameters and DB2 Cloning Tool control statements must match, the following JCL includes sample DB2 Cloning Tool control statements.

The RENAME step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

      //??????? JOB , 'RENAME', CLASS=A, MSGCLASS=X
      //S0      EXEC PGM=IDCAMS
      //SYSPRINT DD  SYSOUT=*
      //SYSIN   DD  *
1     DEL CKZ.WRK.VOLDUMP
1     DEL CKZ.WRK.BCSRECS
2     //S1      EXEC PGM=CKZ00010, REGION=8M
3     //STEPLIB DD  DSN=HLQ?.SCKZLOAD, DISP=SHR
4     //CKZINI  DD  DSN=HLQ?.SCKZPARM(CKZINI), DISP=SHR
5     //SORTMSG DD  SYSOUT=*
6     //SORTWK01 DD UNIT=SYSALLDA, SPACE=(CYL,(10,10))
      //SORTWK02 DD UNIT=SYSALLDA, SPACE=(CYL,(10,10))
      //SORTWK03 DD UNIT=SYSALLDA, SPACE=(CYL,(10,10))
      //SORTWK04 DD UNIT=SYSALLDA, SPACE=(CYL,(10,10))
      //SORTWK05 DD UNIT=SYSALLDA, SPACE=(CYL,(10,10))
      //SORTWK06 DD UNIT=SYSALLDA, SPACE=(CYL,(10,10))
7     //DRSTATS DD  SYSOUT=*
8     //CKZPRINT DD  SYSOUT=*
      //SYSUDUMP DD  SYSOUT=*
9     //JOURNAL DD  DSN=CKZ.JRNL, DISP=OLD
10    //BCSRECS DD  DSN=CKZ.WRK.BCSRECS, UNIT=SYSALLDA, DISP=(,CATLG),
      //          SPACE=(CYL,(10,10))
11    //VOLDUMP DD  DSN=CKZ.WRK.VOLDUMP, UNIT=SYSALLDA, DISP=(,CATLG),
      //          SPACE=(CYL,(10,10))
      //CKKZIN   DD  *
      RENAME
11     SAFE
11     VOLBKUP-DDN(VOLDUMP)

```

```

          MAX-TASKS(5)                -
9        JOURNAL-DDN(JOURNAL)         -
          NOTRENAMED(DELETE,RC(4))    -
          MISSINGCAT(DELETE,RC(8))    -
          ORPHANCATENTRY(KEEP,RC(8))  -
          RECATALOG(N)                -
          DATACLAS(TGTDATA)          -
          MGMTCLAS(TGTMGMT)           -
          STORCLAS-PAIRS(SRCSTOR1,TGTSTOR1 -
                        SRCSTOR2,TGTSTOR2 -
                        DEFAULT_IF_NO_MATCH,TGTSTORX) -
          RENAME-MASKS(               -
            ASRC.** ATGT.**            -
            BSRC.** BTGT.**           -
            PROD.** TEST.**            -
            )

```

/**

1. Deletion of volume dumps and BCSRECS in anticipation of allocating new for each execution. Because these data sets recover target volume information and remove BCS entries during a rerun of the RENAME step, DO NOT set up the rerun JCL to delete these data sets.
2. Execution of DB2 Cloning Tool main program.
3. DB2 Cloning Tool LOAD library must be authorized.
4. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the DB2 Cloning Tool programs.
5. DD for sort messages.
6. DD for sort messages.
7. DD for DRSTATS, SAFE option Dump and Restore output.
8. DD for CKZPRINT output.
9. Journal data set used by DB2 Cloning Tool commands. This VSAM data set is used to pass information between DB2 Cloning Tool steps, and to log information derived from the step executions. The journal file is allocated and cataloged in the COPY step, and used as input and output by the RENAME step. The specified data set name must match the data set allocated in the COPY step. If multiple DB2 Cloning Tool processes are created for different applications, each must use a unique journal data set. In the sample JCL, the control statement JOURNAL-DDN(JOURNAL) specifies that a DD with the name JOURNAL is being used, rather than a data set name.
10. DD for BCSRECS. The data set pointed at by this DD is used to store data set names used if RENAME is rerun, and for the BCSCLEAN command. Required regardless of whether rerunning is anticipated or BCSCLEAN is expected to be used. The data set specified in the RENAME step must match the name used in the corresponding BCSCLEAN step. However, if multiple DB2 Cloning Tool processes are created for different applications, each must have a unique BCSRECS data set.

Note: This file cannot be striped.

11. VOLDUMP DD statement. This DD points to a data set used to back up information on target volumes, in case a rerun of the RENAME step is required. This data set is only used if the SAFE parameter is specified. The normal and abnormal disposition must be CATLG. In the sample JCL, the control statement VOLBKUP-DDN(VOLDUMP) specifies that a DD with the name VOLDUMP is being used rather than a data set name. If the number of volumes involved is high, and the VTOCs, VVDSs, and VTOC indexes are large, this data set may require a substantial amount of space.

Note: The RENAME output also includes output for each volume processed with dynamically allocated DDnames of VVOLSERX.

RESTORE-FROM-DUMPTAPES

This command is not required. RESTORE-FROM-DUMPTAPES allows DB2 Cloning Tool to use the backups on tape from a DB2 BACKUP SYSTEM DUMP utility as the source for cloning.

This command is used as part of the process to clone from DB2 BACKUP SYSTEM DUMP backups on tape. With this process, backup tapes are "restored" to the target volumes instead of the original source volumes. A detailed description of this cloning process can be found in "Cloning scenarios" on page 535.

This command works together with the DB2GETBACKINFO command to clone directly from DB2 system level backup on tape, even when disks are not shared between the two systems. Run DB2GETBACKINFO (with the USE-DUMPTAPES keyword) against the source DB2 system to create the BACKINFO metadata file that describes the dump tape files. Then use the backinfo file as input to RESTORE-FROM-DUMPTAPES. It is not necessary that the target volumes be online at the source DB2 system.

You can specify the target volumes directly by providing a volume list or indirectly by specifying target storage groups, as follows:

- Use only TO-VOLSER to define the target volumes. The volumes are matched against the source volumes and the dump tape volumes in the backinfo file, and assigned in the order they appear in the list.
- Use the TO-STORAGEGROUP and SOURCE-STORAGEGROUP keywords to match sources and targets one-for-one within the specified storage groups.

In either case, to prevent an attempt to restore a larger volume to a smaller one, DB2 Cloning Tool uses volume capacity as a final matching strategy.

During the restore, ADRDSSU changes the volser of the target volume to the volser of the source volume and takes the target volume offline. The volume is relabeled to the target volser, and brought back online. If you specify VARY-SCOPE(GLOBAL) (the default), DB2 Cloning Tool issues sysplex-wide VARY commands to take the volume offline before the restore and brings it back online after the restore. Use VARY-SCOPE(LOCAL) to issue the sysplex-wide VARY commands yourself.

RESTORE-FROM-DUMPTAPES command syntax

RESTORE-FROM-DUMPTAPES

Required Keywords:

```
{ TO-VOLSER (volser1 | volmask1...[ , volsern | volmaskn ] )  
  | TO-STORAGEGROUP( storgrp1...[ , storgrpn ] ) }
```

Optional keywords:

```
BACKINFO-DDN( ddname )  
EXCLUDE-TO-VOLSER(volser1 | volmask1...[ , volsern | volmaskn ] )  
MAX-TAPEDRIVES( nn )  
RERUN  
SIMULATE  
SOURCE-STORAGEGROUP ( sourcestorgrp1...[ , sourcestorgrpn ] )  
STATUS-DDN( ddname )
```

TAPE-UNIT(*name*)
VOLPAIRS-DDN(*ddname*)
USERCATALOGS(*sourcecat1 targetcat1,...[, sourcecatn targetcatn]*)
USERCATALOGS-DDN(*ddname*)
VARY-SCOPE(GLOBAL | LOCAL)

RESTORE-FROM-DUMPTAPES command and keyword definitions

Required keywords are described first, followed by optional keywords.

RESTORE-FROM-DUMPTAPES

Optional command to allow for cloning using the backup tapes from a DB2 BACKUP SYSTEM DUMP backup.

- Required: No
- Restrictions: None

TO-VOLSER (*volser1 | volmask1...[, volsern | volmaskn]*)

This parameter specifies the target volumes that are to be paired with dump volumes.

- Default: None.
- Required: No, but if not supplied, TO-STORAGEGROUP is required.
- Restrictions: Mutually exclusive with TO-STORAGEGROUP.

TO-STORAGEGROUP (*storgrp1...[, storgrpm]*)

This parameter specifies that the target volumes that are required to pair with dump volumes are to be selected from one or more SMS storage groups. All volumes from the specified storage groups are target candidates, except volumes that are excluded via the EXCLUDE-TO-VOLSER parameter. Storage groups are matched one-for-one with SOURCE-STORAGEGROUP storage groups. For more information, see the SOURCE-STORAGEGROUP keyword description.

- Default: None.
- Required: No, but if not supplied, TO-VOLSER is required.
- Restrictions: Mutually exclusive with TO-VOLSER.

BACKINFO-DDN(*ddname*)

This parameter specifies the DD name that points to a file containing the backinfo data. The file named in the DD must have an LRECL of 80.

- Default: None.
- Required: No, unless the BACKINFO DD is not supplied in the JCL.
- Restrictions: None.

EXCLUDE-TO-VOLSER (*volser1 | volmask1...[, volsern | volmaskn]*)

This parameter specifies the volumes to be excluded (not selected as targets) from either the TO-VOLSER parameter or the TO-STORAGEGROUP parameter. After target volumes are excluded, if there are fewer target volumes than dump volumes, the command fails with a return code of 8.

- Default: None.
- Required: No.
- Restrictions: None.

MAX-TAPEDRIVES (*nm*)

This parameter specifies the maximum available number of tape drives available for tape allocation (and implicitly, the number of restore

subtasks). A higher value reduces elapsed run time, but may impact system resources. Valid values are 1-16, or specify an asterisk (*) to allow the RESTORE-FROM-DUMPTAPES command to set the maximum (set to 4 for this release).

- Default: 2.
- Required: No.
- Restrictions: None.

RERUN

This parameter can be used to resume processing from a previous run of the RESTORE-FROM-DUMPTAPES command that uses the same backinfo file. The contents of the status file from the previous run are used to determine the RESTORE-FROM-DUMPTAPES tasks that completed successfully, the tasks that had not started, and the tasks that were in progress. The last known state of each individual restore task, including clip and vary processing, are used to determine the steps that remain.

To use RERUN, you must:

1. Provide the STATUS DD statement on the RESTORE-FROM-DUMPTAPES command, using either a new data set with DISP=NEW, or an existing empty data set that uses the required DCB characteristics.
2. Run the job containing the RESTORE-FROM-DUMPTAPES command. If the command completes with a return code of 0, the status file is no longer needed and can be deleted. If the return code is 8, determine and resolve the error before rerunning the job.
3. To rerun the job, add the RERUN keyword and change the STATUS DD statement to DISP=OLD. Remove all other DD parameters on the STATUS DD except for the DSN, and remove any automatic delete processing. Then submit the job.

- Default: None.
- Required: No.
- Restrictions: None.

SIMULATE

This parameter verifies the syntax, determines the volumes to be processed, and displays the action that will be taken, but does not change the contents of volumes or restore the dump tapes.

- Default: None.
- Required: No.
- Restrictions: None.

SOURCE-STORAGEGROUP (*sourcestorgrp1* ... [, *sourcestorgrpn*])

This parameter specifies the source SMS storage groups. These storage groups are grouped one-for-one with the storage groups named in TO-STORAGEGROUP. Source volumes are paired to target volumes according to these one-for-one storage group mappings; then dump volumes are paired to target volumes accordingly.

- Default: None.
- Required: No.
- Restrictions: None.

STATUS-DDN (*ddname*)

This parameter specifies the ddname of the data set that will be created on a normal (non-RERUN) run and holds the status information of the restore tasks during the job step. It must be retained after a failure (any job step

that ends with return code of 8 or higher) in order to allow a RERUN of the job step. This data set will be used when RERUN is specified to determine which restores need to be done. For more information, see the RERUN parameter description.

- Default: None.
- Required: No; however, a RERUN of the job cannot be attempted unless this parameter or the STATUS DD is supplied.
- Restrictions: None.

TAPE-UNIT(*name*)

This parameter specifies the unit name to use for allocating tape devices for the dump volumes.

- Default: 3490.
- Required: No.
- Restrictions: None.

USERCATALOGS (*sourcecat1 targetcat1, ..., [sourcecatn targetcatn]*)

This parameter specifies the source catalogs that contain data sets from source volumes, and the corresponding target catalog for renamed volume data sets. The specified source catalogs must also be in the backinfo data set in a UCAT record; otherwise, the command fails with a return code of 8.

- Default: None.
- Required: No.
- Restrictions: Can only be specified if the backinfo data set contains UCAT records.

USERCATALOGS-DDN (*ddname*)

This parameter specifies the DD name that points to a file where the user catalog information will be written. The file named in the DD must have an LRECL of 80.

- Default: None.
- Required: No, unless USERCATALOGS is specified and the UCATS DD is not supplied in the JCL.
- Restrictions: None.

VARY-SCOPE (GLOBAL | LOCAL)

When running in a sysplex, specify GLOBAL or LOCAL to determine the scope of VARY commands for target volumes. If you specify GLOBAL, commands are issued across the sysplex. If you specify LOCAL, commands are issued only on the local system (the same system that the job is running on). If you specify LOCAL, resource conflicts might affect the running of the RESTORE-FROM-DUMPTAPES job, other jobs on other systems, or both.

- Default: GLOBAL.
- Required: No.
- Restrictions: None.

VOLPAIRS-DDN (*ddname*)

This parameter specifies the DD name that points to a file where the volume pairs information will be written. The file named in the DD must have an LRECL of 80.

- Default: None.
- Required: No, unless the VOLPAIRS DD is not supplied in the JCL.

- Restrictions: None.

RESTORE-FROM-DUMPTAPES step JCL example

This topic contains an example of RESTORE-FROM-DUMPTAPES step JCL. Sample JCL can be found in the installation library SCKZJCL in member CKZRSTDT.

For completeness and to illustrate where JCL parameters and DB2 Cloning Tool control statements must match, the following JCL includes sample DB2 Cloning Tool control statements. Two sample sets of control cards are provided in the sample JCL.

The RESTORE-FROM-DUMPTAPES step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

      //??????? JOB , 'CKZ RESTORE-F-DT',CLASS=A,MSGCLASS=X,NOTIFY=&SYSUID
      //S0      EXEC PGM=IDCAMS
      //SYSPRINT DD SYSOUT=*
      //SYSIN   DD *
1     DEL HLQ?.WRK.VOLPAIRS
2     DEL HLQ?.WRK.STATUS
3     DEL HLQ?.WRK.UCATS
      SET MAXCC=0
4 //S1      EXEC PGM=CKZ00010,REGION=8M
5 //STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
6 //CKZINI  DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
7 //CKZPRINT DD SYSOUT=*
      //SYSUDUMP DD SYSOUT=*
      //ABNLIGNR DD DUMMY                DO NOT REMOVE IF USING ABENDAID
8 //BACKINFO DD DISP=SHR,DSN=HLQ?.WRK.BACKINFO
9 //VOLPAIRS DD DSN=HLQ?.WRK.VOLPAIRS,
      //      DISP=(,CATLG),UNIT=SYSALLDA,
      //      SPACE=(CYL,(1,1))
10 //STATUS  DD DSN=HLQ?.WRK.STATUS,
      //      DISP=(,CATLG),SPACE=(CYL,(1,1)),
      //      RECOG=KS,KEYLEN=64,KEYOFF=0,
      //      LRECL=600,UNIT=SYSALLDA
11 //UCATS   DD DSN=HLQ?.WRK.UCATS,
      //      DISP=(,CATLG),UNIT=SYSALLDA,
      //      SPACE=(CYL,(1,1))
      //CKZIN  DD *
      //*****
      /* SAMPLE CONTROL STATEMENTS
      //*****
      /* SAMPLE 1: RESTORE FROM DUMPTAPES USING UP TO 4 TAPE DRIVES,
      /*      USING SOURCE VOLUMES FROM SOURCE-STORAGEGROUP,
      /*      AND THEIR MATCHING TAPES IN THE BACKINFO FILE,
      /*      PAIR WITH TARGET VOLUMES FROM TO-STORAGEGROUP
      /*
      /* RESTORE-FROM-DUMPTAPES           -
12 /* MAX-TAPEDRIVES ( 4 )              -
8  /* BACKINFO-DDN( BACKINFO           ) -
9  /* VOLPAIRS-DDN( VOLPAIRS           ) -
11 /* USERCATALOGS-DDN( UCATS           ) -
13 /* SOURCE-STORAGEGROUP (            -
      /* SGRP1 SGRP2                    -
      /* )                                -
14 /* TO-STORAGEGROUP (                -
      /* SGRPY1 SGRPY2                  -
      /* )                                -
15 /* USERCATALOGS (                   -
      /* USERCAT.SRC01 USERCAT.TGT01   -
      /* USERCAT.SRC02 USERCAT.TGT02   -

```

```

|          )
|          /**
|          /** SAMPLE 2: RESTORE FROM DUMPTAPES USING UP TO 6 TAPE DRIVES,
|          /**          USING ALL SOURCE VOLUMES AND MATCHING TAPES
|          /**          IN THE BACKINFO FILE, PAIR WITH TARGET VOLUMES
|          /**          FROM TO-VOLSER
|          /**
|          /** RESTORE-FROM-DUMPTAPES          -
| 12 /** MAX-TAPEDRIVES ( 6 )          -
| 8  /** BACKINFO-DDN( BACKINFO          ) -
| 9  /** VOLPAIRS-DDN( VOLPAIRS          ) -
| 11 /** USERCATALOGS-DDN( UCATS          ) -
| 16 /** TO-VOLSER (          -
|          /**          TGT001 TGT002 TGT003          -
|          /**          )          -
| 15 /** USERCATALOGS (          -
|          /**          USERCAT.SRC01 USERCAT.TGT01          -
|          /**          USERCAT.SRC02 USERCAT.TGT02          -
|          /**          )
|          /**
|          /**

```

1. Deletion of volume pairs data set in anticipation of allocating new for each execution.
2. Deletion of the status data set in anticipation of allocating new for each execution.
3. Deletion of user catalog data set in anticipation of allocating new for each execution.
4. Execution of DB2 Cloning Tool main program.
5. DB2 Cloning Tool LOAD library must be authorized.
6. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the DB2 Cloning Tool programs.
7. DD for CKZPRINT output.
8. The backinfo data set that is created by the DB2 Cloning Tool DB2GETBACKINFO command from the source DB2 system. It contains information about the DB2 system level backup information from HSM LIST COPYPOOL, and will be used as input to the RESTORE-FROM-DUMPTAPES command.
9. Output DD for volume pairing information for use in later COPY(PGM(NONE)) step.
10. DD to record the status and progress of RESTORE-FROM-DUMPTAPES command. This DD is required for RERUN.
11. Output DD for user catalog information for use in later COPY(PGM(NONE)) step.
12. Specifies the maximum available number of tape drives available for tape allocation (and implicitly, the number of restore subtasks).
13. Specifies the source SMS storage groups.
14. Specifies the target SMS storage groups that will be used to pair the target volumes with the dump volumes. Storage groups are matched one-for-one with the SOURCE-STORAGEGROUP storage groups.
15. Lists the source catalogs that contain data sets from source volumes, and the corresponding target catalog for renamed volume data sets.
16. Specifies the target volumes that are to be paired with dump volumes.

UCATOPTIONS

This command is not required. UCATOPTIONS BACKUP will back up source catalogs when the COPY command used the USERCATALOGS-NOBACKUP keyword. If VOLSER was specified for the source catalog when executing COPY with the USERCATALOGS-NOBACKUP keyword, the backup is taken from the copies of the source catalogs that reside on target volumes. UCATOPTIONS BACKUP must be run before the RENAME command.

UCATOPTIONS LIST will list the source and target ICF catalog pairs and CATWORK data set names in the DB2 Cloning Tool journal.

UCATOPTIONS UPDATE will allow the target user catalog name(s) or CATWORK data set name(s) to be changed.

For example, if you specify a target catalog(s) in the COPY command, and want to change the target catalog(s) used during the RENAME command, this command will list the source and target ICF catalog pairs currently in the journal and allow you to update the journal with the new target ICF catalog names.

Attention: If the COPY command used the USERCATALOGS-NOBACKUP keyword, the catalog backup can be done by using the UCATOPTIONS command with the BACKUP keyword. However, the source ICF catalog information being backed up needs to be in synchronization with the contents of the copied volumes. If UCATOPTIONS BACKUP is used (meaning that USERCATALOGS-NOBACKUP was specified for the COPY command), the user must ensure that the source ICF catalogs are logically at the same point in time as when the volumes were copied.

UCATOPTIONS command syntax

UCATOPTIONS

Required keywords:

{ JOURNAL-DSN(*data set name*) | JOURNAL-DDN(*ddname*) }

Optional keywords:

BACKUP [FORCE] | LIST | UPDATE

If UPDATE is specified, one of the following keywords must be specified:

NEWCATWORKS(*current-dsn1*, *new-dsn1*, *current-dsn2*, *new-dsn2*, ...) |
NEWCATWORKS-DDN(*ddname*)
NEWTARGETS(*srcusercatalog1*, *newtgtusercatalog1*,
 srcusercatalog2, *newtgtusercatalog2*, ...) |
NEWTARGETS-DDN(*ddname*)

UCATOPTIONS command and keyword definitions

Required keywords are described first, followed by optional keywords.

UCATOPTIONS

Optional command that will either list the user catalog pairs from the DB2 Cloning Tool journal, or allow the target user catalog name(s) to be changed, or back up source catalogs.

- Required: No
- Restrictions: None

JOURNAL-DSN (*data set name*)
or **JOURNAL-DDN** (*ddname*)

This parameter supplies either the data set name of the DB2 Cloning Tool journal file, or a ddname assumed via the JCL to point to a journal data set. The journal is used to pass information between DB2 Cloning Tool steps.

- Default: None
- Required: Yes
- Restrictions: None

BACKUP [FORCE] | LIST | UPDATE

BACKUP requests that the source catalogs be backed up. The backup was not done by COPY because the USERCATALOGS-NOBACKUP keyword was used. If VOLSER was specified for the source catalog when executing COPY with the USERCATALOGS-NOBACKUP keyword, the backup is taken from copies of the source catalogs that reside on target volumes.

UCATOPTIONS BACKUP must be run before the RENAME command. If VOLPAIRSDEVN-NOCLIP or VOLPAIRSDEVN-NOCLIP-DDN were used with USERCATALOGS-NOBACKUP in the COPY, VOLOPTIONS OFFLINECLIP must be run before UCATOPTIONS BACKUP which must be run before the RENAME command.

BACKUP FORCE requests that the source catalogs be backed up even if they have already been backed up.

LIST requests a display of the current source and target ICF catalog pairs.

UPDATE changes the existing target ICF catalog entries found in the journal with the new target ICF catalog names. UPDATE can not be used if the catalog backup has not been done.

Note: The existing source ICF catalogs must be entered in pairs with the new target ICF catalogs as shown in the NEWTARGETS and NEWTARGETSDDN keyword description that follows.

- Default: LIST
- Required: No
- Restrictions: None for LIST or BACKUP. UPDATE requires one of the following: NEWCATWORKS, NEWCATWORKS-DDN, NEWTARGETS, or NEWTARGETSDDN.

NEWCATWORKS (*current-dsn1, new-dsn1, current-dsn2, new-dsn2, ...*)
or **NEWCATWORKS-DDN**(*ddname*)

NEWCATWORKS specifies the current CATWORK data set name and the new catwork data set name. The current CATWORK data set name must match an DB2 Cloning Tool journal entry.

new-dsn1 will replace the current CATWORK data set name with the value of *current-dsn1*, with the new CATWORK data set name in the journal entry.

NEWCATWORKS-DDN specifies a DD name assumed via the JCL to point to a data set containing the current and new CATWORK data set name pairs. The pairs are the same format as in the NEWCATWORKS keyword.

- Default: None
- Required: No
- Restrictions: May only be specified with UPDATE.

NEWTARGETS(*srcusercatalog1*, *newtgtusercatalog1*, *srcusercatalog2*,
newtgtusercatalog2, ...)

or **NEWTARGETS-DDN**(*ddname*)

NEWTARGETS specifies the current source ICF catalog and the new target ICF catalog. The current source ICF catalog must match an DB2 Cloning Tool journal entry.

newtgtusercatalog1 will replace the current target user catalog which is paired with *srcusercatalog1* in the journal entry.

NEWTARGETS-DDN specifies a DD name assumed via the JCL to point to a data set containing the source and target ICF catalog pairs. The pairs are the same format as in the **NEWTARGETS** keyword.

- Default: None
- Required: No
- Restrictions: May only be specified with UPDATE.

UCATOPTIONS step JCL example

| This topic contains an example of UCATOPTIONS step JCL. Sample JCL can be
| found in the installation library SCKZJCL in member CKZUCATO.

For completeness and to illustrate where JCL parameters and DB2 Cloning Tool control statements must match, the following JCL includes sample DB2 Cloning Tool control statements.

The UCATOPTIONS step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```
1 //??????? JOB , 'UCATOPTIONS', CLASS=A, MSGCLASS=X  
2 //S1 EXEC PGM=CKZ00010, REGION=8M  
3 //STEPLIB DD DSN=HLQ?.SCKZLOAD, DISP=SHR  
4 //CKZINI DD DSN=HLQ?.SCKZPARM(CKZINI), DISP=SHR  
5 //CKZPRINT DD SYSOUT=*  
6 //SYSUDUMP DD SYSOUT=*  
7 //JOURNAL DD DSN=CKZ.JRNL, DISP=OLD  
8 //CKZIN DD *  
9 UCATOPTIONS  
10 JOURNAL-DDN(JOURNAL)  
11 /*
```

1. Execution of DB2 Cloning Tool main program.
2. DB2 Cloning Tool LOAD library must be authorized.
3. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the DB2 Cloning Tool programs.
4. DD for CKZPRINT output.
5. Journal data set used by DB2 Cloning Tool commands. This VSAM data set is used to pass information between DB2 Cloning Tool steps, and to log information derived from the step executions. The journal file is allocated and cataloged in the COPY step and used as input to the UCATOPTIONS command. In the sample JCL, the control statement JOURNAL-DDN(JOURNAL) specifies that a DD with the name JOURNAL is being used rather than a data set name.

VARYOFF

This command is not required. VARYOFF is provided to vary target or source volumes offline on only the local system where the command is run, on all systems in the sysplex, or on all systems in the sysplex other than the local system.

Note: VARYOFF should not be used in JES3 environment. The VARYOFF command issues an MVS VARY command, which is not recommended for JES3-managed devices.

The volumes to be varied offline can be specified by:

- Using the vary file. The volumes are determined when the VARYOFF command is run by reading a provided COPY command (COPY-CMD-DDN(COPYCMD) DD that points to a file which contains DB2 Cloning Tool COPY command syntax. The COPY syntax is read for volume information but is not executed during the VARYOFF processing). The volumes are then saved in the vary file for use by a subsequent VARYON command to bring the volumes online.
The vary file can be used when the volumes have different device numbers on different systems. For this case the VARYOFF command would be run with SCOPE(LOCAL) on each system with a separate vary file for each system. See the example using the vary off file in the topic “VARYOFF step JCL example” on page 461.
- Using an existing journal data set. The volumes have been predetermined by the DB2 Cloning Tool COPY command that created the journal and can be used if the volumes will be varied offline later in the process.

When using SCOPE(SYSPLEX(ALL)) or SCOPE(SYSPLEX(OTHER)) the volumes need to have the same device numbers on the systems in the sysplex.

VARYOFF command syntax

VARYOFF

Required keywords:

```
{ JOURNAL-DSN( data set name ) | JOURNAL-DDN( ddname ) |  
  VARY-DDN( ddname ) }  
TARGET | SOURCE
```

Required only if VARY-DDN(ddname) is specified:

```
COPY-CMD-DDN( ddname )
```

Optional keywords:

```
MAX-VOLS-PER-CMD( nn | 8 )  
SCOPE( LOCAL | SYSPLEX( { ALL | OTHER } [ , T( nnn ) ] ) )  
SIMULATE  
VOL-ALREADY-OFFLINE( { QUIT | CONTINUE } [ , ( RC( nnn | 8 ) ] )
```

VARYOFF command and keyword definitions

Required keywords are described first, followed by optional keywords.

VARYOFF

Optional command to vary volumes offline.

- Required: No
- Restrictions: None

JOURNAL-DSN (*data set name*)
or **JOURNAL-DDN** (*ddname*)

This parameter supplies either the data set name of the DB2 Cloning Tool journal file, or a DD name assumed via the JCL to point to a journal data set. The journal is used to pass information between DB2 Cloning Tool steps.

- Default: None
- Required: Yes, if VARY-DDN is not specified.
- Restrictions: Mutually exclusive with VARY-DDN.
- Short form(s): JDSN, JDDN

TARGET | SOURCE

Specifies the volume set to use. **TARGET** specifies that the target volumes will be used. **SOURCE** specifies that the source volumes will be used.

- Default: None
- Required: Yes
- Restrictions: None

VARY-DDN(*ddname*)

This parameter supplies the DD name of the DB2 Cloning Tool vary file assumed via the JCL to point at a vary data set.

If multiple DB2 Cloning Tool setups are used for different volume groups, DO NOT use the same vary data set. Each DB2 Cloning Tool volume group needs a different vary data set.

The vary file is used to pass VOLSERs and device numbers between an DB2 Cloning Tool VARYOFF step and an DB2 Cloning Tool VARYON step. Therefore, as noted in the JCL comments, it must be cataloged in the VARYOFF step when first created, and referenced in the VARYON step.

- Default: None
- Required: Yes, if JOURNAL-DSN and JOURNAL-DDN are not specified.
- Restrictions: Mutually exclusive with JOURNAL-DSN and JOURNAL-DDN.

COPY-CMD-DDN(*ddname*)

This parameter supplies the DD name of the DB2 Cloning Tool copy command.

- Default: None
- Required: Yes, if VARY-DDN is specified.
- Restrictions: None

MAX-VOLS-PER-CMD(*nm* | **8**)

Specifies the maximum number of volumes that will be used in a single SYSPLEX vary offline command.

- Default: 8
- Required: no
- Restrictions: Used only when SCOPE(SYSPLEX(..)) is specified.

SCOPE(LOCAL | **SYSPLEX**({ **ALL** | **OTHER** } [, **T**(*nmn*)]))

Specifies the scope of the vary offline commands.

LOCAL specifies that the vary offline commands will be issued only on the local system.

SYSPLEX specifies that the vary offline commands will be issued sysplex-wide.

ALL specifies that the volumes will be varied offline to the local system and vary offline commands will be routed to all the other systems in the sysplex.

OTHER specifies that vary offline commands will be routed to all the other systems in the sysplex, no vary offlines will be issued on the local system.

T specifies an optional timeout interval that will be added to the sysplex vary commands.

- Default: Local
- Required: No
- Restrictions: None

SIMULATE

Simulate will verify the syntax and determine the volumes to be varied offline but will not issue any vary offline commands.

- Default: None
- Required: No
- Restrictions: None
- Short form: SIM

VOL-ALREADY-OFFLINE({ QUIT | CONTINUE } [, (RC(*nnn* | 8)])

Specifies the action to be taken when a volume to be processed is already offline.

QUIT specifies that processing will quit when the first volume already offline is encountered.

CONTINUE specifies that processing will continue with the next volume when a volume already offline is encountered.

RC specifies the return code that will be used when a volume already offline is encountered.

- Default: QUIT, RC(8)
- Required: No
- Restrictions: Used only when SCOPE(LOCAL) or SCOPE(SYSPLEX(ALL)) are specified.

VARYOFF step JCL example

This topic contains several examples of VARYOFF step JCL.

VARYOFF step JCL example - varying volumes offline using vary file

For completeness and to illustrate where JCL parameters and DB2 Cloning Tool control statements must match, the following JCL includes sample DB2 Cloning Tool control statements. Sample JCL for this example can be found in the installation library SCKZJCL in member CKZVOFFV.

The VARYOFF step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

      /??????? JOB  , 'VARYOFF', CLASS=A, MSGCLASS=X
      //S0      EXEC PGM=IDCAMS
      //SYSPRINT DD  SYSOUT=*
      //SYSIN    DD  *
1     DEL CKZ.VARY
      SET MAXCC=0
2     //S1      EXEC PGM=CKZ00010, REGION=6M
3     //STEPLIB DD  DSN=HLQ?.SCKZLOAD, DISP=SHR
4     //CKZINI  DD  DSN=HLQ?.SCKZPARAM(CKZINI), DISP=SHR
5     //CKZPRINT DD  SYSOUT=*
      //SYSUDUMP DD  SYSOUT=*
6     //VARY    DD  DSN=CKZ.VARY, RECOG=KS, KEYLEN=64, KEYOFF=0,
      //          DISP=(,CATLG), UNIT=SYSALLDA,
      //          LRECL=600, SPACE=(CYL,(10,10))
      //CKZIN   DD  *
          VARYOFF                -
7     TARGET                    -
8     SCOPE(LOCAL)              -
9     COPY-CMD-DDN(COPYCMD)    -
6     VARY-DDN(VARY)
      //*
9     //COPYCMD DD  *
          COPY                    -
          DATA-MOVER(           -
              COPYCMDLIMIT(24)  -
          )                       -
          FROM-VOLSER(VSRC02)    -
7     TO-VOLSER(VTGT02)        -
          USERCATALOGS(         -
              USERCAT.SRC01 USERCAT.TGT01 -
              USERCAT.SRC02 USERCAT.TGT02 -
          )                       -
          CATWORK-DSN(CKZ.WRK.* ) -
          JOURNAL-DDN(JOURNAL)
      //*

```

- 1
1. Deletion of vary data set in anticipation of allocating new for each execution.
 2. Execution of DB2 Cloning Tool main program.
 3. DB2 Cloning Tool LOAD library must be authorized.
 4. DD for CKZINI, SCKZPARAM member. The CKZINI member of the HLQ?.SCKZPARAM library provides variables to the DB2 Cloning Tool programs.
 5. DD for CKZPRINT output.
 6. Vary data set. This VSAM data set is used to pass information between the DB2 Cloning Tool VARYOFF and VARYON steps. The vary file is allocated and cataloged in the VARYOFF step and used as input to the VARYON command. In the sample JCL, the control statement VARY-DDN(VARY) is specifying that a DD with the name VARY is being used.
 7. TARGET parameter that specifies the target volumes found in the journal will be varied offline. In the COPY command, the target volumes specified will be used.
 8. SCOPE parameter specifying LOCAL will cause the vary offline commands to be issued only on the local system.
 9. The COPY command. The COPY command is used to derive the volumes for processing. In the sample JCL, the control statement COPY-CMD-DDN(COPYCMD) is specifying that a DD with the name COPYCMD is being used to read the copy command.

VARYOFF step JCL example - varying volumes offline using journal

For completeness and to illustrate where JCL parameters and DB2 Cloning Tool control statements must match, the following JCL includes sample DB2 Cloning Tool control statements. Sample JCL for this example can be found in the installation library SCKZJCL in member CKZVOFFJ.

The VARYOFF step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

  //??????? JOB  , 'VARYOFF', CLASS=A, MSGCLASS=X
1 //S1      EXEC PGM=CKZ00010, REGION=6M
2 //STEPLIB DD  DSN=HLQ?.SCKZLOAD, DISP=SHR
3 //CKZINI  DD  DSN=HLQ?.SCKZPARAM(CKZINI), DISP=SHR
4 //CKZPRINT DD SYSOUT=*
  //SYSUDUMP DD SYSOUT=*
5 //JOURNAL DD  DSN=CKZ.JRNL, DISP=OLD
  //CKZIN   DD  *
    VARYOFF          -
6   TARGET          -
7   SCOPE(SYSPLEX(OTHER)) -
5   JOURNAL-DDN(JOURNAL)
  //*
```

1. Execution of DB2 Cloning Tool main program.
2. DB2 Cloning Tool LOAD library must be authorized.
3. DD for CKZINI, SCKZPARAM member. The CKZINI member of the HLQ?.SCKZPARAM library provides variables to the DB2 Cloning Tool programs.
4. DD for CKZPRINT output.
5. Journal data set used by DB2 Cloning Tool commands. This VSAM data set is used to pass information between DB2 Cloning Tool steps and to log information derived from the step executions. The journal file is allocated and cataloged in the COPY step and used as input to the VARYOFF command. In the sample JCL, the control statement JOURNAL-DDN(JOURNAL) is specifying that a DD with the name JOURNAL is being used rather than a data set name.
6. TARGET parameter that specifies the target volumes found in the journal will be varied offline
7. SCOPE parameter specifying SYSPLEX(OTHER) will cause the vary offline commands to be routed to all other systems in the sysplex.

VARYON

This command is not required. VARYON is provided to vary target or source volumes online. The VARYON command can vary volumes online on only the local system where the command is run, on all systems in the sysplex, or on all systems in the sysplex other than the local system.

When using SCOPE(SYSPLEX(ALL)) or SCOPE(SYSPLEX(OTHER)) the volumes need to have the same device numbers on the systems in the sysplex.

Note: VARYON should not be used in JES3 environment. The VARYON command issues an MVS VARY command, which is not recommended for JES3-managed devices.

The volumes to be varied online can be specified by:

- Using the vary file. By using the vary file, the volumes have been predetermined by the DB2 Cloning Tool VARYOFF command that created the vary file. The vary file can be used when the volumes have different device numbers on different systems. For this case, the VARYON command would be run with SCOPE(LOCAL) on each system with a separate vary file for each system.
- Using an existing journal data set. The volumes have been predetermined by the CKZ COPY command that created the journal.

When using SCOPE(SYSPLEX(ALL)) or SCOPE(SYSPLEX(OTHER)) the volumes need to have the same device numbers on the systems in the sysplex.

VARYON command syntax

VARYON

Required keywords:

```
{ JOURNAL-DSN( data set name ) | JOURNAL-DDN( ddname ) |
  VARY-DDN( ddname ) }
```

Required only if JOURNAL-DSN(data set name) or JOURNAL-DDN(ddname) is specified:

```
TARGET | SOURCE
```

Optional keywords:

```
MAX-VOLS-PER-CMD( nn | 8 )
SCOPE( LOCAL | SYSPLEX( { ALL | OTHER } [ , T( nnn ) ] ) )
SIMULATE
VOL-ALREADY-ONLINE( { QUIT | CONTINUE } [ , ( RC( nnn | 8 ) ) ] )
WRONG-VOLSER( { QUIT | CONTINUE } [ , ( RC( nnn | 8 ) ) ] )
```

VARYON command and keyword definitions

Required keywords are described first, followed by optional keywords.

VARYON

Optional command to vary volumes online.

- Required: No
- Restrictions: None

JOURNAL-DSN (data set name) or JOURNAL-DDN (ddname)

This parameter supplies either the data set name of the DB2 Cloning Tool journal file, or a DD name assumed via the JCL to point to a journal data set. The journal is used to pass information between DB2 Cloning Tool steps.

- Default: None
- Required: Yes, if VARY-DDN is not specified.
- Restrictions: Mutually exclusive with VARY-DDN.
- Short form(s): JDSN, JDDN

TARGET | SOURCE

Specifies the volume set to use. **TARGET** specifies that the target volumes will be used. **SOURCE** specifies that the source volumes will be used.

- Default: None
- Required: Yes, if JOURNAL-DSN or JOURNAL-DDN is specified.
- Restrictions: Mutually exclusive with VARY-DDN.

VARY-DDN(*ddname*)

This parameter supplies the DD name of the DB2 Cloning Tool vary file assumed via the JCL to point at a vary data set.

If multiple DB2 Cloning Tool setups are used for different volume groups, DO NOT use the same vary data set. Each DB2 Cloning Tool volume group needs a different vary data set.

The vary file is used to pass VOLSERs and device numbers between an DB2 Cloning Tool VARYOFF step and an DB2 Cloning Tool VARYON step. Therefore, as noted in the JCL comments, it must be cataloged in the VARYOFF step when first created, and referenced in the VARYON step.

- Default: None
- Required: Yes, if JOURNAL-DSN and JOURNAL-DDN are not specified.
- Restrictions: Mutually exclusive with JOURNAL-DSN and JOURNAL-DDN.

MAX-VOLS-PER-CMD(*nm* | 8)

Specifies the maximum number of volumes that will be used in a single SYSPLEX vary online command.

- Default: 8
- Required: no
- Restrictions: Used only when SCOPE(SYSPLEX(..)) is specified.

SCOPE(LOCAL | SYSPLEX({ ALL | OTHER } [, T(*nnn*)]))

Specifies the scope of the vary online commands.

LOCAL specifies that the vary online commands will be issued only on the local system.

SYSPLEX specifies that the vary online commands will be issued sysplex-wide.

ALL specifies that the volumes will be varied online to the local system and vary online commands will be routed to all the other systems in the sysplex.

OTHER specifies that vary online commands will be routed to all the other systems in the sysplex, no vary online commands will be issued on the local system.

T specifies an optional timeout interval that will be added to the sysplex vary commands.

- Default: Local
- Required: No
- Restrictions: None

SIMULATE

Simulate will verify the syntax and determine the volumes to be varied offline but will not issue any vary online commands.

- Default: None
- Required: No
- Restrictions: None
- Short form: SIM

VOL-ALREADY-ONLINE({ QUIT | CONTINUE } [, (RC(*nnn* | 8))])

Specifies the action to be taken when a volume to be processed is already online.

QUIT specifies that processing will quit when the first volume already offline is encountered.

CONTINUE specifies that processing will continue with the next volume when a volume already online is encountered.

RC specifies the return code that will be used when a volume already online is encountered.

- Default: QUIT, RC(8)
- Required: No
- Restrictions: Used only when SCOPE(LOCAL) or SCOPE(SYSPLEX(ALL)) are specified.

WRONG-VOLSER({ QUIT | CONTINUE } [, (RC(*nnn* | 8)])

Specifies the action to be taken when a volume that is brought online has a different VOLSER than expected.

QUIT specifies that processing will quit when the first volume with a wrong VOLSER is encountered.

CONTINUE specifies that processing will continue with the next volume when a volume with a wrong VOLSER is encountered.

RC specifies the return code that will be used when a volume with a wrong VOLSER is encountered.

- Default: QUIT, RC(8)
- Required: No
- Restrictions: Used only when SCOPE(LOCAL) or SCOPE(SYSPLEX(ALL)) are specified.

VARYON step JCL example

This topic contains several examples of VARYON step JCL.

VARYON Step JCL – Example to vary volumes online using vary file

For completeness and to illustrate where JCL parameters and DB2 Cloning Tool control statements must match, the following JCL includes sample DB2 Cloning Tool control statements. Sample JCL for this example can be found in the installation library SCKZJCL in member CKZVONV.

The VARYON step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

  //??????? JOB  , 'VARYON', CLASS=A, MSGCLASS=X
1 //S1      EXEC PGM=CKZ00010, REGION=6M
2 //STEPLIB DD  DSN=HLQ?.SCKZLOAD, DISP=SHR
3 //CKZINI  DD  DSN=HLQ?.SCKZPARM(CKZINI), DISP=SHR
4 //CKZPRINT DD SYSOUT=*
  //SYSUDUMP DD SYSOUT=*
5 //VARY    DD  DSN=CKZ.VARY, DISP=OLD
  //CKZIN   DD  *
  VARYON
6   SCOPE(LOCAL)
5   VARY-DDN(VARY)
  //*
```

1. Execution of DB2 Cloning Tool main program.

2. DB2 Cloning Tool LOAD library must be authorized.
3. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the DB2 Cloning Tool programs.
4. DD for CKZPRINT output.
5. Vary data set. This VSAM data set is used to pass information between the DB2 Cloning Tool VARYOFF and VARYON steps. The vary file is allocated and cataloged in the VARYOFF step and used as input to the VARYON command. In the sample JCL, the control statement VARY-DDN(VARY) is specifying that a DD with the name VARY is being used.
6. SCOPE parameter specifying LOCAL will cause the vary online commands to be issued only on the local system.

VARYON Step JCL – Example to vary volumes online using journal

For completeness and to illustrate where JCL parameters and DB2 Cloning Tool control statements must match, the following JCL includes sample DB2 Cloning Tool control statements. Sample JCL for this example can be found in the installation library SCKZJCL in member CKZVONJ.

For completeness and to illustrate where JCL parameters and DB2 Cloning Tool control statements must match, the following JCL includes sample DB2 Cloning Tool control statements.

The VARYON step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

//??????? JOB  , 'VARYON', CLASS=A, MSGCLASS=X
1 //S1      EXEC PGM=CKZ00010, REGION=6M
2 //STEPLIB DD DSN=HLQ?.SCKZLOAD, DISP=SHR
3 //CKZINI  DD DSN=HLQ?.SCKZPARM(CKZINI), DISP=SHR
4 //CKZPRINT DD SYSOUT=*
  //SYSUDUMP DD SYSOUT=*
5 //JOURNAL DD DSN=CKZ.JRNL, DISP=OLD
  //CKZIN   DD *
    VARYON          -
6   TARGET          -
7   SCOPE(SYSPLEX(OTHER)) -
5   JOURNAL-DDN(JOURNAL)
  //*
```

1. Execution of DB2 Cloning Tool main program.
2. DB2 Cloning Tool LOAD library must be authorized.
3. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the DB2 Cloning Tool programs.
4. DD for CKZPRINT output.
5. Journal data set used by DB2 Cloning Tool commands. This VSAM data set is used to pass information between DB2 Cloning Tool steps, and to log information derived from the step executions. The journal file is allocated and cataloged in the COPY step and used as input to the VARYON command. In the sample JCL, the control statement JOURNAL-DDN(JOURNAL) is specifying that a DD with the name JOURNAL is being used rather than a data set name.
6. TARGET parameter that specifies the target volumes found in the journal will be varied online.

7. SCOPE parameter specifying SYSPLEX(OTHER) will cause the vary online commands to be routed to all other systems in the sysplex.

VOLOPTIONS

This command is not required. VOLOPTIONS is intended for situations where the DB2 Cloning Tool COPY step is run at one site (SITEA) and the DB2 Cloning Tool RENAME step is run at another site (SITEB). VOLOPTIONS LIST, CLIP, and UPDATE are intended to support such scenarios.

VOLOPTONS can also be used when the target volumes are offline to the DB2 Cloning Tool COPY step and it is not desirable for DB2 Cloning Tool COPY to re-label and vary online the target volumes. For instance, the target offline volumes will be backed up prior to the DB2 Cloning Tool RENAME step being run. For this function, COPY would use the VOLPAIRSDEVN-NOCLIP or VOLPAIRSDEVN-NOCLIP-DDN keywords, which cause COPY to not re-label or vary online the target volumes. The offline target volumes are then backed up, and VOLOPTIONS OFFLINECLIP is used to re-label and vary online the target volumes prior to running the RENAME command.

VOLOPTONS can also be used when the target volumes are online and it is desired to vary them offline and relabel them to their corresponding source volume volser. VOLOPTIONS UNCLIP can be used to support this scenario.

For example: SITEA runs the DB2 Cloning Tool COPY which FlashCopys (or snaps) source volumes SRC001, SRC002, and SRC003 to target volumes TGT001, TGT002, and TGT003. SITEA then dumps TGT001, TGT002, TGT003 to tape and sends them to SITEB. The COPY also backs up the source user catalogs and establishes the source/target user catalog relationship.

In addition, SITEA backs up the DB2 Cloning Tool journal file and the data sets created under the CATWORK-DSN prefix. The journal file and CATWORK-DSN data sets are created by the DB2 Cloning Tool COPY command.

SITEB restores the DB2 Cloning Tool journal file and the CATWORK-DSN data sets on their system. The CATWORK-DSN data set names must match those used by SITEA. The journal file and CATWORK-DSN data sets must be cataloged.

SITEB restores the tape(s) using 'ADRDSSU RESTORE COPYVOLID'.

At this point, SITEB has SRC001, SRC002, and SRC003 online to their system. The data sets on these volumes are not cataloged.

In order for DB2 Cloning Tool RENAME to rename and catalog the SITEB data sets, the volumes need to be clipped to the corresponding target volume serials.

VOLOPTIONS CLIP will do this function. It will vary SRC001, SRC002, and SRC003 offline on the image where the command is executed. It will then use ICKDSF to change SRC001 to TGT001, SRC002 to TGT002, and SRC003 to TGT003. The target volumes will be varied online by VOLOPTIONS CLIP to the image where the command is executed.

If SITEB needs to see the volume pairs used by SITEA, VOLOPTIONS LIST will display the current source/target volume pairs.

DB2 Cloning Tool uses the DB2 Cloning Tool journal to identify and communicate source and target volume pairs to DB2 Cloning Tool commands. If SITEB needs to use different target volume serials than those used at SITEA, VOLOPTIONS UPDATE NEWTARGETS will allow the current target volume serials in the DB2 Cloning Tool journal to be changed.

VOLOPTIONS LIST could be used to provide the initial input for the NEWTARGETS keyword. The new target volume serial(s) would need to be added to the source/target pairs from the VOLOPTIONS LIST command.

VOLOPTIONS UPDATE NEWTARGETS and NEWTARGETSDEVN do not support changing any target volume serial to a source volume serial.

VOLOPTIONS command syntax

VOLOPTIONS

Required keywords:

{ JOURNAL-DSN(*data set name*) | JOURNAL-DDN(*ddname*) }

Optional keywords:

LIST | CLIP | OFFLINECLIP | UNCLIP | UPDATE
RESUME
SIMULATE

Required only if UPDATE is specified:

{ NEWTARGETS(*srcvolser1 tgtvolser1 newtgtvolser1 ... [, srcvolsern
tgtvolsern newtgtvolsern]*) |
NEWTARGETS-DDN(*ddname*) |
NEWTARGETSDEVN
(*srcvolser1 tgtvolser1 newtgtvolser1 newtgtdevn1 ... [, srcvolsern tgtvolsern
newtgtvolsern newtgtdevnn]*) |
NEWTARGETSDEVN-DDN(*ddname*) }

VOLOPTIONS command and keyword definitions

Required keywords are described first, followed by optional keywords.

VOLOPTIONS

Optional command to use when the COPY command is run at one site and the RENAME command is run at another.

- Required: No
- Restrictions: None

JOURNAL-DSN (*data set name*)

or **JOURNAL-DDN** (*ddname*)

This parameter supplies either the data set name of the DB2 Cloning Tool journal file, or a DD name assumed via the JCL to point at a journal data set.

If multiple CKZ setups are used for different volume groups, DO NOT use the same journal data set. Each CKZ 'application' needs a different journal data set.

The journal is used to pass information between CKZ steps (e.g., from the COPY step to the RENAME step). Therefore, as noted in the JCL comments, it must be cataloged in the COPY step when first created, and referenced as OLD in subsequent steps.

Do not delete the data set in the last step, in case restarts or reruns need journal information. It is preferable to delete and replace the data set only at the fresh start of a COPY.

- Default: None
- Required: Yes
- Restrictions: None
- Short form(s): JDSN, JDDN

LIST | CLIP | OFFLINECLIP | UNCLIP | UPDATE

LIST requests a display of the current source volume serial/target volume serial pairs.

CLIP requests that the online source volume serials be changed with ICKDSF to their paired target volume serials.

OFFLINECLIP requests that the offline target volumes which currently have source volume serials have their volume serials changed with ICKDSF to their target volume serials and varied online. If VOLPAIRSDEVN-NOCLIP or VOLPAIRSDEVN-NOCLIP-DDN was used with USERCATALOGS-NOBACKUP in the COPY, VOLOPTIONS OFFLINECLIP must be run before UCATOPTIONS BACKUP which must be run before the RENAME command.

UNCLIP requests that the online target volumes be varied offline and the target serials be changed with ICKDSF to their paired source volume serials.

UPDATE requests changing specified target volume serials to new target volume serials.

- Default: LIST
- Required: No
- Restrictions: UPDATE requires NEWTARGETS or NEWTARGETSDEVN. No restrictions for LIST, CLIP, or OFFLINECLIP, or UNCLIP.

RESUME

RESUME specifies that CLIP, OFFLINECLIP, or UNCLIP should resume processing of any volumes that failed to be completely processed by a previous CLIP, OFFLINECLIP, or UNCLIP command.

Prior to running CLIP, OFFLINECLIP, or UNCLIP with RESUME, the problem that caused a volume to not be completely processed by CLIP, OFFLINECLIP, or UNCLIP should be resolved.

- Default: None
- Required: No
- Restrictions: This keyword can only be specified if CLIP, OFFLINECLIP, or UNCLIP is specified.

SIMULATE

Simulate will verify the syntax and determine the volumes to be processed and will display what action would have been taken but will not vary any volumes offline or online or change any volume serials with ICKDSF or update the journal.

- Default: None
- Required: No
- Restrictions: None
- Short form: SIM

NEWTARGETS (*srcvolser1 tgtvolser1 newtgtvolser1 ... [, srcvolsern tgtvolsern newtgtvolsern]*)

or NEWTARGETS-DDN (*ddname*) | **NEWTARGETSDEVN** (*srcvolser1 tgtvolser1 newtgtvolser1 newtgtdevn1 ... [srcvolsern tgtvolsern newtgtvolsern newtgtdevnn]*)

or NEWTARGETSDEVN-DDN (*ddname*)

NEWTARGETS specifies the current source volume serial, the current target volume serial, and the new target volume serial. The current source volume serial and current target volume serial must match the CKZ journal entries. For example: 'newtgtvolser1' will replace 'tgtvolser1' in the journal entry. The short form is NTGT.

NEWTARGETS-DDN specifies a DD name which has a file containing the volume serial triplets. The triplets are the same format as in the **NEWTARGETS** keyword.

NEWTARGETSDEVN specifies the current source volume serial, the current target volume serial, and the new target volume serial and new target device number. The current source volume serial and current target volume serial must match the CKZ journal entries. For example: 'newtgtvolser1' will replace 'tgtvolser1' and 'newtgtdevn1' will replace the target device number in the journal entry. The short form is NTGTD.

NEWTARGETSDEVN-DDN specifies a DD name which has a file containing the volume serial and devn quadruplets. The quadruplets are the same format as in the **NEWTARGETSDEVN** keyword.

- Default: None
- Required: One of the **NEWTARGETS** keywords is required if **UPDATE** is specified.
- Restrictions: May only be specified with **UPDATE**.

VOLOPTIONS step JCL example

This topic contains an example of **VOLOPTIONS** step JCL. Sample JCL can be found in the installation library **SCKZJCL** in member **CKZVOLOP**.

For completeness and to illustrate where JCL parameters and DB2 Cloning Tool control statements must match, the following JCL includes sample DB2 Cloning Tool control statements.

The **VOLOPTIONS** step JCL is shown in the following figure. The numbers in the first column are not part of the JCL, but correspond to notes following the sample JCL that contain further information.

```

    /??????? JOB , 'VOLOPTIONS', CLASS=A, MSGCLASS=X
1 //S1      EXEC PGM=CKZ00010, REGION=8M
2 //STEPLIB DD DSN=HLQ?.SCKZLOAD, DISP=SHR
3 //CKZINI  DD DSN=HLQ?.SCKZPARM(CKZINI), DISP=SHR
4 //CKZPRINT DD SYSOUT=*
  //SYSUDUMP DD SYSOUT=*
5 //JOURNAL DD DSN=CKZ.JRNL, DISP=OLD
  //CKZIN   DD *
    VOLOPTIONS
5     JOURNAL-DDN(JOURNAL)
  //*
```

1. Execution of DB2 Cloning Tool main program.
2. DB2 Cloning Tool **LOAD** library must be authorized.

3. DD for CKZINI, SCKZPARM member. The CKZINI member of the HLQ?.SCKZPARM library provides variables to the DB2 Cloning Tool programs.
4. DD for CKZPRINT output.
5. Journal data set used by DB2 Cloning Tool commands. This VSAM data set is used to pass information between DB2 Cloning Tool steps, and to log information derived from the step executions. The journal file is allocated and cataloged in the COPY step and used as input to the VOLOPTIONS command. In the sample JCL, the control statement JOURNAL-DDN(JOURNAL) specifies that a DD with the name JOURNAL is being used rather than a data set name.

The previous JCL example used LIST, the default, so no keyword was specified. The following two examples are provided for CLIP and UPDATE keywords:

Example using VOLOPTIONS (CLIP):

```
VOLOPTIONS CLIP -
    JOURNAL-DDN(JOURNAL)
```

Example using VOLOPTIONS (UPDATE):

```
VOLOPTIONS UPDATE
    NEWTARGETS(VSRC01,VTGT11,VTGT01 -
               VSRC02,VTGT22,VTGT02 -
               VSRC03,VTGT13,VTGT03) -
    JOURNAL-DDN(JOURNAL)
```

Chapter 24. DB2 Cloning Tool Table Space Cloning commands

All DB2 Cloning Tool Table Space Cloning commands are invoked by executing the main program, CKZ00500. Functionality is selected by specifying the appropriate DB2 Cloning Tool Table Space Cloning command and parameters.

Most DB2 Cloning Tool Table Space Cloning commands support and modify the behavior of the COPY command.

The DB2 Cloning Tool Table Space Cloning commands are listed in the following table and also explained in detail in separate topics.

Table 59. DB2 Cloning Tool Table Space Cloning commands overview

DB2 Cloning Tool Table Space Cloning Command	Description
"COPY"	Controls all phases of replicating the DB2 table spaces and index spaces that are to be cloned.
"DATASUBTYPE" on page 510	Optional command that may be used to specify the data subtype value (BIT, MIXED, or SBCS) needed by data masking.
"HLQDDDF" on page 511	Optional command allows DB2 Cloning Tool Table Space Cloning to pass input and output DDs to ADRDSSU or the EMC API.
"LISTDEF" on page 512	Used to select to source DB2 table spaces and index spaces to be copied to the target DB2.
"SET" on page 516	Specifies the local DB2 subsystem for the source job and the TCP/IP server job.

COPY

The COPY command is required. COPY invokes FlashCopy or SnapShot to perform data set copies if the DATA-MOVER(PGM(ADRDSSU)) is specified, or invokes data set copies via TimeFinder/Clone Mainframe Snap Facility's data set level support facility if the DATA-MOVER(PGM(EMCAPI)) is specified, or if DATA-MOVER(PGM(NONE)) is specified COPY assumes copies have been created by you. In all cases, COPY captures the DB2 catalog data pertaining to source and target data sets.

What the COPY command does:

1. Connects to the source DB2 catalog specified by the LOCAL-SSID keyword of the SET command
2. Reads the source DB2 catalog to get a list of table spaces and index spaces specified by the LISTDEF DD CKZLSTDF
3. Acquires the attributes and IDs of the source DB2 table spaces and index spaces remaining after LISTDEF processing is complete
4. Builds VSAM data set names from the entries in the source DB2 catalog

5. Queries the target DB2 catalog, specified by the COPY command sub-keyword SSID of keyword TARGET-DB2, and verifies the existence of matching target DB2 table spaces and index spaces
6. Acquires from the target DB2 catalog the attributes and IDs of the existing target DB2 table spaces and index spaces
7. Issues DB2 STOP commands for the source and the target (if requested) if PGM=ADRDSSU or EMCAPI
8. Invokes FlashCopy or SnapShot (if available) if PGM=ADRDSSU, or TimeFinder/Clone if PGM=EMCAPI
9. Issues DB2 START commands for the source if PGM=ADRDSSU or EMCAPI
10. Outputs target commands to the SYNCDB2 data set for input to the target job

COPY command syntax

COPY

Required Keywords:

TARGET-DB2(SSID(*sourcesubsystem*))
 conditionally required keywords . . .
 optional keywords . . .

Conditionally Required Keywords:

TARGET-DB2(LOC(*sourcelocation*)
 IP(*ppp.ppp.ppp.ppp*)
 DEFVCAT(*targethlq*)
 [USR(*ddfuserid*)PASSWORD(*ddfpassword*)]
 PORT(*port*)

Optional Keywords:

[ALLOW-COPY-ON-MISMATCH(Y | N)]
 [ALWAYS-COPY-HISTORY-TABLES (Y | N)]
 [ALWAYS-COPY-INDEXSPACES (Y | N)]
 [AUTO-START-SOURCE-SPACE(Y | N | R)]
 [AUTO-START-TARGET-SPACE(Y | N)]
 [AUTO-STOP-TARGET-SPACE(Y | N)]
 [CATALOG-PREFETCH(
 [ENABLE-PREFETCH (Y | N)]
 [ENABLE-SOURCE-PREFETCH(Y | N)]
 [ENABLE-TARGET-PREFETCH (Y | N)]
 [SOURCE-PREFETCH-DATABASE-LIST(*sourcedatabase1*, *sourcedatabase2*,
 . . . *sourcedatabasen*)]
 [TARGET-PREFETCH-DATABASE-LIST (*targetdatabase1*, *targetdatabase2*,
 . . . *targetdatabasen*)]]
 [CHECK-DATASET-COMPATIBILITY(Y | N)]
 [CHECK-INDEX-KEYS(Y | N)]
 [COPY-IF-NO-DB2-TARGET-OBJECTS(Y | N)]
 [COPY-IJ-TO-NONEXISTENT-TARGET(Y | N)]
 [DATA-MASKING (Y | N)]
 [DATA-MOVER(PGM(ADRDSSU | EMCAPI | SRCIMCPY | NONE)
 [,CMDDNAME(*ddname*)]
 [,FASTREP(PREF | REQ | NONE)]
 [,FCTOPPRCPRIARY [(PRESMIRREQ | PRESMIRPREF | PRESMIRNONE)]
 [,NULLSTORCLASS]]
 [DATASETS-TO-COPY-DDN(*ddname*)]
 [DSNS-PER-COPY (*numberofdatasets*)]
 [DSS-COPY-COMMANDS (*numberofcommands*)]
 [EXTEND-TARGET-PBG-TABLESPACE(Y | N)]
 [FUZZY-COPY(Y | N)]
 [IDCAMS-DDN(*ddname*)]
 [IGNORE-RF-MISMATCH-IF-NO-VAR-COLS(Y | N)]
 [INCLUDE-ALL-RI (Y | N)]

```

[ JOB-TEMPLATE( inddname1, outddname1,...[ , inddnamen, outddnamen ] ) ]
[ LOG-APPLY( LA-ENABLE( Y | N )
  [ ,COMMON-CONSISTENT-POINT( Y | N ) ]
  [ ,DATA-SHARING-MEMBERS( id | ssid | zparm ) ]
  [ ,END-POINT( TO_CURRENT | TO_QUIESCE | TO_LOGPOINT X'byte_string' ) ]
  [ ,MINILOG-HLQ( miniloghighlevelqualifier ) ]
  [ ,MINILOG-LARGE-FILE-TYPE( Y | N ) ]
  [ ,MINILOG-UNIT-TYPE( SYSALLDA ) ]
  [ ,MINILOG-QUANTITY-IN-TRACKS( Y | N ) ]
  [ ,MINILOG-PRIMARY-QUANTITY( nnnn ) ]
  [ ,MINILOG-SECONDARY-QUANTITY( nnnn ) ]
  [ ,MINILOG-VOLUME-COUNT( nnn ) ]
  [ ,MINILOG-DATACLAS( dataclas ) ]
  [ ,MINILOG-STORCLAS( storclas ) ]
  [ ,MINILOG-MGMTCLAS( mgmtclas ) ]
  [ ,NUMBER-OF-BUFFERS( nn ) ]
  [ ,NUMBER-OF-CHANNEL-PROGRAMS( nn ) ]
  [ ,QUIESCE-POINT( Y | N ) ]
  [ ,REBUILD-COPY-NO-INDEXES( Y | N ) ]
  [ ,SORTFILE-LARGE-FILE-TYPE( Y | N ) ]
  [ ,SORTFILE-UNIT-TYPE( SYSALLDA ) ]
  [ ,SORTFILE-PRIMARY-QUANTITY( nnnn ) ]
  [ ,SORTFILE-SECONDARY-QUANTITY( nnnn ) ]
  [ ,SORTFILE-VOLUME-COUNT( nnn ) ]
  [ ,SORTFILE-DATACLAS( dataclas ) ]
  [ ,SORTFILE-STORCLAS( storclas ) ]
  [ ,SORTFILE-MGMTCLAS( mgmtclas ) ]
  [ ,SORT-PROGRAM( DFSORT | SYNCSORT ) ]
  [ ,SPACES-PER-MINILOG( numberofspaces ) ]
  [ ,UNIFIED-WARNING( Y | N ) ]
  [ ,USE-TCPIP( Y | N ) ]
  [ ,WARN-IF-SKIP-QUIESCE( Y | N ) ]
  [ ,WARN-IF-TS-DEFINED-LOG-NO( Y | N ) ]
  [ ,WORKFILE-LARGE-FILE-TYPE( Y | N ) ]
  [ ,WORKFILE-UNIT-TYPE( SYSALLDA ) ]
  [ ,WORKFILE-QUANTITY-IN-TRACKS( Y | N ) ]
  [ ,WORKFILE-PRIMARY-QUANTITY( nnnn ) ]
  [ ,WORKFILE-SECONDARY-QUANTITY( nnnn ) ]
  [ ,WORKFILE-VOLUME-COUNT( nnn ) ]
  [ ,WORKFILE-DATACLAS( dataclas ) ]
  [ ,WORKFILE-STORCLAS( storclas ) ]
  [ ,WORKFILE-MGMTCLAS( mgmtclas ) ]
  [ ,ZPARAM-MEMBER( zparammembername ) ] ) ]
[ LONGVAR-COMPATIBILITY( Y | N ) ]
[ OBJECT-MISMATCH-RETURN-CODE( mismatchtypel,returncode1,
  . . . mismatchtypen,returncoden ) ]
[ OBJECT-TRANSLATE(
  DATABASE, sourcedatabase, targetdatabase
    [DATABASE, sourcedatabasen, targetdatabasen ]
  TABLESPACE, sourcetablespace1, targettablespace1 . . .
    [TABLESPACE, sourcetablespacen, targettablespacen ]
  TABLE, source1, target1 . . .
    [TABLE, source1n, target1n ]
  INDEXSPACE, sourceindexspace1, targetindexspace1 . . .
    [INDEXSPACE, sourceindexspacen, targetindexspacen ]
  INDEX, sourceindex1, targetindex1 . . .
    [INDEX, sourceindexn, targetindexn ]
  CREATOR, sourcecreator1, targetcreator1 . . .
    [CREATOR, sourcecreatorn, targetcreatorn ] )
  VCAT, sourcevcat1, targetvcat1 . . .
    [VCAT, sourcevcatn, targetvcatn ] ) ]
[ PROCESS-DDL( DDL-ENABLE( Y | N )
  [ ,PROCESS-TYPE( [ Y | N | G | X | A ] ) ]
  [ ,PROCESS-DDL-DDN( ddname ) ]
  [ ,IGNORE-CREATE-OBJECT-EXISTS( Y | N ) ]
  [ ,GENERATE-DDL-DEFAULTS( Y | N ) ]
  [ ,DDL-ATTRIBUTE-CHANGE

```

```

|          (attributename,srcvalue,trgvalue,applytotype,applytoobject) ] ) ]
| [ REPLACE-TARGET-DSN( Y | N ) ]
| [ RESET-LOGRBA( Y | N ) ]
| [ SIMULATE( A | N | Y ) ]
| [ SLOUT-DDN( ddname ) ]
| [ START-SOURCE-DDN( ddname ) ]
| [ STOP-SOURCE-DDN( ddname ) ]
| [ STOP-TARGET-DDN( ddname ) ]
| [ SYNCDB2-DDN( ddname ) ]
| [ TARGET-JOB-INDEX-REBUILD-DDN( ddname ) ]
| [ TARGET-JOB-REPAIR-DDN( ddname ) ]
| [ V7-MIGRATED-OBJECTS-PRESENT( Y | N ) ]
| [ WARN-IF-OBJECT-NOT-TRANSLATED( Y | N ) ]
| [ WARN-ON-DATASET-EXTENSION-MISMATCH( Y | N ) ]
| [ WARN-ON-INCOMPLETE-RI( Y | N ) ]
| [ WARN-ON-SIMPLE-TABLESPACE( Y | N ) ]
| [ WARN-ON-VERSIONS( Y | N ) ]
| [ XMLSTRING-DDN( ddname ) ]

```

COPY command and keyword definitions

Required keywords are described first, followed by optional keywords.

Required COPY keywords

COPY The COPY command controls all phases of replicating the DB2 table spaces and index spaces that are to be cloned.

- Default: None
- Required: Yes
- Restrictions: None

**TARGET-DB2 DEFVCAT(targethlq) IP(ppp.ppp.ppp.ppp) LOC(sourcelocation)
PASSWORD(dfpassword) PORT(port) SSID(sourcesubsystem) USR(dfuserid)**

The TARGET-DB2 parameter specifies the target DB2 subsystem and supplies the parameters needed to access the target via DDF or TCP/IP. Also, a default high level qualifier is supplied for the target DB2 data sets.

SSID - the subsystem ID of the target DB2.

LOC - the DDF location name of the target DB2.

IP - the IP address (IPv4 or IPv6) of the LPAR on which the target DB2 subsystem resides.

DEFVCAT - the default High Level Qualifier used for the target data sets if the target DB2 table spaces and index spaces are not pre-defined.

USR - an optional user ID that may be needed to access DDF.

PASSWORD - an optional password that may be needed to access DDF.

PORT - the port that the TCP/IP server uses to wait for requests from the source job and that the source job uses to connect to the TCP/IP server job. The port assigned to the TCP/IP server cannot be a port used by DB2 or any other application.

- Default: None
- Required:
 - SSID is an unconditionally required parm.
 - LOC is required if DDF needed to connect to the target subsystem.
 - IP is required if TCP/IP used to connect to the target subsystem.
 - DEFVCAT is required if one or more source table spaces or index spaces do not exist on the target subsystem.

- USR and PASSWORD may be required for DDF connection to the target subsystem.
- PORT is required to use a different port than the one specified in PARMLIB.
- Restrictions: None
- Short form: TDB2

Optional COPY keywords

ALLOW-COPY-ON-MISMATCH (Y | N)

The ALLOW-COPY-ON-MISMATCH parameter specifies whether a copy should be allowed if one or more object attributes do not match between the source and target table spaces or index spaces.

N – The table space or index space is not copied and the job completes with a return code of 4.

Y – The copy is still performed if the CKZINI PARMLIB parameter MAX_RC is set to 4 (or if the CKZINI PARMLIB parameter MAX_RC is set to 0 but MAX-RC on the SET command is set to 4).

Note: When this keyword is set to N, set the PARMLIB keywords MAX_COPY_RC to 8 and the MAX_RC to 4 (or use the SET command keywords MAX-COPY-RC(8) and MAX-RC(4)) to allow DB2 Cloning Tool Table Space Cloning to copy only table spaces and index spaces that are safe to copy, such as those not in restrictive or advisory states or those without object mismatches (such as column length or type). For all other source table spaces and index spaces, error messages will be produced and the table spaces and index spaces are not copied.

- Default: N
- Required: No
- Restrictions: None
- Short form: ACOM

ALWAYS-COPY-HISTORY-TABLES (Y | N)

ALWAYS-COPY-HISTORY-TABLES can be used to select history tables associated with all objects specified on a LISTDEF. To select both history tables and non-history tables, set ALWAYS-COPY-HISTORY-TABLES to Y. When ALWAYS-COPY-HISTORY-TABLES is set to N, history tables are not selected; however, history tables can be selected by using the HISTORY parameter on a LISTDEF command.

The following examples show how to combine ALWAYS-COPY-HISTORY-TABLES and the HISTORY parameter on a LISTDEF to select history tables.

Example: Include only non-history tables in a table space:

```
INCLUDE TABLESPACES TABLESPACE table_space_name ALL
ALWAYS-COPY-INDEXSPACES(N)
ALWAYS-COPY-HISTORY-TABLES(N)
```

Example: Include only history tables with all history indexes in a table space:

```
INCLUDE TABLESPACES TABLESPACE table_space_name HISTORY ALL
ALWAYS-COPY-INDEXSPACES(Y)
ALWAYS-COPY-HISTORY-TABLES(N)
```

Example: Include both history and non-history tables with all indexes in a database.

```
INCLUDE TABLESPACES DATABASE database_name ALL  
ALWAYS-COPY-INDEXSPACES(Y)  
ALWAYS-COPY-HISTORY-TABLES(Y)
```

- Default: N
- Required: No
- Restrictions: Valid only for DB2 10 and later subsystems.
- Short form: ACHTB

ALWAYS-COPY-INDEXSPACES (Y | N)

When set to Y, the ALWAYS-COPY-INDEXSPACES parameter specifies that for every table space included in a LISTDEF, all index spaces are also included. No INCLUDE INDEXSPACES syntax is required in the LISTDEF.

- Default: N
- Required: No
- Restrictions: None
- Short form: CPYIX

AUTO-START-SOURCE-SPACE (Y | N | R)

The AUTO-START-SOURCE-SPACE parameter allows the source job to optionally start each source DB2 table space and index space after the copy process is complete.

Y – (the default) DB2 Cloning Tool Table Space Cloning starts the source table spaces and index spaces in RW mode after the copy is complete.

N - source table spaces and index spaces are left stopped after the copy is complete.

R– the source table spaces and index spaces are restored to the status they were before executing the source job; however, if a source table space or index space has an initial status of STOP or STOPP, the table space or index space is started in RW.

If you set this parameter to Y or R, and a table space or index space has a status that is not STOP, RW, RO or STOPP, a RC of 8 is issued and the table space or index space is not copied.

If FUZZY-COPY(Y) has been specified, AUTO-START-SOURCE-SPACE is ignored.

- Default: Y
- Required: No
- Restrictions: Valid only when source table spaces and index spaces have a status of STOP, RW, RO, or STOPP; source table spaces and index spaces with any other status will not be copied by DB2 Cloning Tool Table Space Cloning.
- Short form: ASTRSS

AUTO-START-TARGET-SPACE(Y | N)

The AUTO-START-TARGET-SPACE parameter is passed to the DB2 Cloning Tool Table Space Cloning target job and allows the target job to optionally start each target DB2 table space and index space after the cloning process is complete. The target space is not started after the source job completes.

Refer to “How DB2 Cloning Tool Table Space Cloning starts and stops table spaces and index spaces” on page 156 for more information about stopping and starting target table spaces and index spaces.

- Default: Directed by CKZINI token AUTO_START_TARGET_SPACE in :DSN_COPY_OPTIONS section. If no token value, then Y.
- Required: No
- Restrictions: None
- Short form: ASTRTS

AUTO-STOP-TARGET-SPACE(Y | N)

The AUTO-STOP-TARGET-SPACE parameter allows the source job to optionally stop the target DB2 table spaces and index spaces. If the target table spaces and index spaces are already stopped, set this to N. This will prevent the DB2 DISPLAY command from being issued against all the target table spaces and index spaces.

Refer to “How DB2 Cloning Tool Table Space Cloning starts and stops table spaces and index spaces” on page 156 for more information about stopping and starting target table spaces and index spaces.

- Default: Directed by CKZINI token AUTO_STOP_TARGET_SPACE in :DSN_COPY_OPTIONS section. If no token value, then Y.
- Required: No
- Restrictions: None
- Short form: ASTPTS

CATALOG-PREFETCH (

ENABLE-PREFETCH (Y | N)

ENABLE-SOURCE-PREFETCH (Y | N)

ENABLE-TARGET-PREFETCH (Y | N)

SOURCE-PREFETCH-DATABASE-LIST (*sourcedatabase1, sourcedatabase2, . . . sourcedatabasen*)

TARGET-PREFETCH-DATABASE-LIST (*targetdatabase1, targetdatabase2, . . . targetdatabasen*)

) This parameter can be entered on the COPY command in the source job. For additional information, refer to the topic “Considerations for using catalog prefetch to populate the object cache” on page 163.

- Default: None
- Required: No
- Restrictions: This command is not available when running in DB2 Version 8 Compatibility Mode.
- Short form: CATPF

ENABLE-PREFETCH - Y enables the other prefetch commands.

- Default: N
- Required: No
- Restrictions:

When using this command, only specify objects using the LISTDEF statements INCLUDE TABLESPACES DATABASE or INCLUDE TABLESPACES TABLESPACE, even if SOURCE-PREFETCH -DATABASE-LIST is in use. In addition, if TABLESPACE object specification is used, it must be a base table space; that is, LOB and XML table spaces cannot be specified. The LOB specification must be ALL; LOB, XML and blank may not be specified. If these restrictions are not acceptable, specify

ENABLE-PREFETCH(N) or ENABLE-SOURCE-PREFETCH(N) and SELECT using WHERE clauses will be used to read the source and/or target catalogs.

- Short form: ENAPP

ENABLE-SOURCE-PREFETCH - Y enables prefetch for the source catalog. This command allows the source objects from one or more databases to be saved in memory (cached) during a single pass of the catalog tables. If this parameter is set to Y in the source job and the SOURCE-PREFETCH-DATABASE-LIST exists, it will be used. If SOURCE-PREFETCH-DATABASE-LIST does not exist, LISTDEF will be used to find objects to prefetch.

- Default: N
- Required: No
- Restrictions: None
- Short form: ENASPF

ENABLE-TARGET-PREFETCH - Y enables prefetch for the target catalog. This command allows the target objects from one or more databases to be saved in memory (cached) during a single pass of the catalog tables. When Y is specified for this parameter, the target objects will be mapped from the source objects. This includes object translation if required. When Y is specified for this parameter and the TCPIP server job is in use, target objects and the enable flag come from the source job.

- Default: N
- Required: No
- Restrictions: None
- Short form: ENATPF

SOURCE-PREFETCH-DATABASE-LIST - Specifies a list of databases to be prefetched when accessing the source catalog. When using LISTDEF with more than one database, multiple reads of the same catalog tables can take place. Multiple reads can be eliminated by using this prefetched list. Duplicate databases are not detected and cause no problems. More than 1,000 databases can be entered.

Important: If all databases are not included in the prefetch database list, the objects will not be found and the source job will not run correctly.

- Default: None.
- Required: No
- Restrictions: None
- Short form: SPFDBL

TARGET-PREFETCH-DATABASE-LIST - Specifies a list of databases to be prefetched when accessing the target catalog. Connection to the target can be CAF, DDF or TCPIP. This subcommand can be used in the source job and is passed to the TCPIP server job if it is in use. More than 1,000 databases can be entered. If this command is not entered, the list of target databases to be cached is generated from the source data set names mapped to target names using object translate. When copying a large number of table spaces, compare source job run times with and without this database list to determine which gives the best performance.

Important: If all databases are not included in the prefetch database list, the objects will not be found and the source job will not run correctly.

- Default: None.
- Required: No
- Restrictions: None
- Short form: TPFDBL

CHECK-DATASET-COMPATIBILITY(Y | N)

This parameter checks several VSAM data set attributes to determine if they are compatible between the source and target subsystems. The following attributes are checked:

- If the data set is in extended format.
- If the data set was allocated using extended addressability.
- Whether the data set can be compressed
- Whether the data set is striped.
- Whether the data set can be spanned.

These attributes must be the same between the source and target subsystems. When one or more data set incompatibilities are found, no copies are attempted and the source job ends with RC=8.

Note:

If the target objects have been created with DEFINE NO, no comparison is performed.

IDCAMS LISTCAT commands are issued for each data set pair (source and target). IDCAMS commands and responses are output to the CKZLOG data set. Use DATA-MOVER(PGM(NONE)) to check data set compatibility. Correct the data sets found to be incompatible. Then use PGM(ADRDSSU) to make the copies.

- Default: N.
- Required: No
- Restrictions: None
- Short form: CKDSN

CHECK-INDEX-KEYS(Y | N)

When set to Y, the CHECK-INDEX-KEYS parameter makes additional checks on index compatibility. LIMITKEY is always checked and is normally sufficient. In some cases, a mismatch is missed unless the keys are read and checked also. If using index caching, run once with CHECK-INDEX-KEYS(Y), change any indexes to correct the mismatches, and then run with CHECK-INDEX-KEYS(N) or the default. Use of this command may cause performance degradation, especially when there are many indexes to copy or caching is being used to access the catalog (caching is not supported for index keys).

- Default: N.
- Required: No
- Restrictions: None
- Short form: CIXKY

COPY-IF-NO-DB2-TARGET-OBJECTS (Y | N)

The COPY-IF-NO-DB2-TARGET-OBJECTS parameter specifies if source VSAM data sets are to be copied even if the target DB2 table spaces and index spaces do not exist.

| Target objects that were created with DEFINE NO are always copied.
| Target objects that were created with DEFINE NO and that do not have
| existing data sets will require additional processing. Refer to
| "Considerations for target objects created using DEFINE NO" on page 155
| for more information.

- Default: Directed by CKZINI token
COPY_IF_NO_DB2_TARGET_OBJECTS in :DSN_COPY_OPTIONS
section. If no token value, then N.
- Required: No
- Restrictions: None
- Short form: CINTO

COPY-IJ-TO-NONEXISTENT-TARGET (Y | N)

| Use this keyword to create data sets on the target subsystem without the
| need to rename and delete temporary data sets on the target. This keyword
| can be used when the target data sets have not yet been created and the
| target objects were not created using DEFINE NO; refer to the topic
| "Considerations for target objects created using DEFINE NO" on page 155.

| When COPY-IJ-TO-NONEXISTENT-TARGET is set to N and no target data
| sets are found (and the objects were not defined using DEFINE NO), the
| data sets are copied from source to target using F0001 as the fifth-node
| qualifier. IDCAMS DELETE and RENAME statements are generated to
| rename the target objects with the correct I or J fifth-node qualifier and
| delete the F0001 data sets.

| When COPY-IJ-TO-NONEXISTENT-TARGET is set to Y, you must also
| specify the target data set VCAT using the OBJECT-TRANSLATE VCAT
| command or the TARGET-DB2 DEFVCAT command. If the target data set
| VCAT is not supplied, the copy will not be performed. The target objects
| are created as follows:

- If the objects exist in the target catalog, the data sets are copied from the
source using the I or J fifth-node qualifier that appears in the target
catalog.
- If the objects do not exist in the target catalog, the data sets are copied
from the source using the source catalog object I or J fifth-node qualifier
for the target data sets.

| No IDCAMS DELETE and RENAME statements are generated, as no
| temporary data sets (F0001 as fifth-node qualifier) are created on the target.

- Default: N
- Required: N
- Restrictions: None
- Short form: CITNT

DATA-MASKING (Y | N)

| If set to Y, the DATA-MASKING keyword enables data masking.

| If you specify DATA-MASKING(Y) for a table, assume that all indexes on
| that table must be rebuilt. Omit the table's indexes from the LISTDEF, and
| ensure ALWAYS-COPY-INDEXSPACES is defaulted (N) or specifies N
| explicitly to prevent all indexes from being included in copy processing.
| After executing the target job, rebuild all the indexes that were omitted
| from the copy.

- Default: N

- Required: N
- Restrictions: DATA-MASKING(Y) is not valid with PGM(SRCIMCPY).
- Short form: MASKING

DATA-MOVER (PGM(ADRDSSU | EMCAPI | SRCIMCPY | NONE)
[,CMDDDDNAME(ddname)]
[,FASTREP(PREF | REQ | NONE)]
[,FCTOPPRCPPRIMARY [(PRESMIRREQ | PRESMIRPREF |
PRESMIRNONE)]
[,NULLSTORCLASS]

The DATA-MOVER parameter specifies the program to be used to initiate copies and copy options.

PGM(ADRDSSU) is the default; it specifies that COPY is to initiate FlashCopy or SnapShot 'under the covers' via execution of DSS.

PGM(EMCAPI) - species that DB2 Cloning Tool Table Space Cloning is to invoke EMC TimeFinder/Clone to make the copies using the data set snap facility. Refer to the topic "GLOBAL command values for EMC TimeFinder/Clone Mainframe Snap Facility data set level support" on page 508 for additional information.

PGM(SRCIMCPY) specifies that image copies are to be used as the input to the table space cloning process. When SRCIMCPY is specified, the cloning process identifies the required image copies based on the specified end point (point in time) of the clone. In most cases, the first full image copy that is older than the specified end point is used, and incremental image copies that were taken after the full image copy but before the end point are merged with the full image copy. LOG-APPLY processing is required and is used to apply the logs to bring objects to a consistent state on the target. Before the data sets are copied to the target, pages are fixed (OBIDs are translated and certain page fields are reset). Objects supported are table spaces, index spaces and LOBs. Image copies can be SHRLEVEL REFERENCE or CHANGE.

PGM(NONE) specifies that no DATA-MOVER is to be invoked by COPY. NONE infers that data set copies will be created by you between the execution of the source job and the execution of the target job. When NONE is specified, COPY still captures necessary DB2 catalog information. NONE may also be used to verify object compatibility from source to target and to ensure parameters are correctly specified.

CMDDDDNAME is used only with job templates. The DD name indicates the name of an output data set that contains a job built using the JOB-TEMPLATE parameter of the COPY command. Note that each step in the externally built set of input cards is a separate call to ADRDSSU. The JCL step execution cards are discarded. The DSS input cards consisting of DSS commands such as COPY, INCLUDE, PARALLEL and data set names are sent to ADRDSSU as is. This parameter is not available for PGM(EMCAPI). CMDDDDNAME is not valid with PGM(SRCIMCPY).

FASTREP (DSS parameter) - indicates whether fast replication is preferred (PREF), required (REQ), or not required (NONE). PREF is the default. DB2 Cloning Tool Table Space Cloning will set up the source/target pairs for a fast replication if PREF or REQ is specified. DB2 Cloning Tool Table Space Cloning will allow a 'normal' copy if NONE is specified. If the level of ADRDSSU indicates it supports this keyword, the keyword will be passed to ADRDSSU. FASTREP is not valid with PGM(SRCIMCPY).

FCTOPPRCPRIARY (DSS parameter) - Indicates that a FlashCopy target volume can also be a PPRC primary volume. This applies to ESS devices only. This does not apply when FASTREP(NONE) is also specified. FCTOPPRCPRIARY is not valid with PGM(SRCIMCPY).

IBM Remote Pair FlashCopy (also known as Preserve Mirror) can be specified by including one of the optional keywords. Preserve Mirror mirrors the FlashCopy command that is issued at the local site to the remote site. This allows FlashCopy operations to occur to PPRC primary volumes without affecting the PPRC duplex state. IBM Remote Pair FlashCopy must be installed in the storage controller along with the corresponding software support in z/OS. In addition, both the source and target volumes being PPRC primary volumes and in the same storage controller and their corresponding PPRC secondary volumes being in the same storage controller.

Specify one of the following to use this functionality:

- PRESMIRREQ: Require the use of Preserve Mirror. If a Preserve Mirror operation cannot be accomplished, the FlashCopy operation will not be completed.
- PRESMIRPREF: Prefer the use of Preserve Mirror. If a Preserve Mirror operation cannot be accomplished, the FlashCopy operation is still performed.
- PRESMIRNONE: Do not use Preserve Mirror.

When any of these options is specified, they will be used as part of the source to target volume pairing criteria. The specified option will also be passed to ADRDSSU as part of the generated copy commands. When FCTOPPRCPRIARY is not specified or if the capability is not supported by the ESS, a PPRC primary volume is not eligible to become a FlashCopy target volume. For additional information about Preserve Mirror, refer to the documentation for your version of z/OS.

Do not specify the FCTOPPRCPRIARY keyword with the FASTREP (NONE) keyword.

NULLSTORCLASS - Use this keyword to pass a null storage class to the automatic class selection (ACS) routine. This allows the ACS routine to assign storage classes to target data sets. For more information on this keyword, consult the DFSMSdss Storage Administration documentation.

DATASETS-TO-COPY-DDN(*ddname*)

The DATASETS-TO-COPY-DDN parameter supplies the *ddname* pointing at the data set which will contain a list of TO and FROM data set names derived from the LISTDEF command input. The data set pointed to by DATASETS-TO-COPY-DDN must have an LRECL of 80 and RECFM of FB.

The purpose of this data set is to assist in replicating the DB2 VSAM objects when using a method other than DB2 Cloning Tool Table Space Cloning.

The data sets to copy and their names are output in CKZPRINT also. This is just a more convenient form for submitting copy commands to a copy product that DB2 Cloning Tool Table Space Cloning does not currently support. Refer to the JOB-TEMPLATE parameter for the preferred method of building copy jobs to be run outside of DB2 Cloning Tool Table Space Cloning.

- Default: None
- Required: No. This is optional even if PGM(NONE).

- Restrictions: Valid only if DATA-MOVER(PGM(NONE)) is specified.
- Short form: DSN-DDN

DSNS-PER-COPY(*numberofdatasets*)

The DSNS-PER-COPY parameter specifies the number of data sets to send to DSS in a single copy command. Valid values are 1 to 255; 255 is the recommended value.

- Default: PARMLIB specification for DSNS_PER_COPY, or 255 if not specified in PARMLIB.
- Required: No.
- Restrictions: None
- Short form: DSNRPC

DSS-COPY-COMMANDS (*numberofcommands*)

The DSS-COPY-COMMANDS parameter specifies the number of DSS copy commands to send to DSS in a single invocation. Valid values are 1 to 256; specifying a large value may result in a storage shortage and S878 abends. When cloning a large number of data sets, the default of 24 might be too high.

- Default: PARMLIB specification for DSS_COPY_COMMANDS, or 24 if not specified in PARMLIB.
- Required: No.
- Restrictions: None
- Short form: DSSCMD5

EXTEND-TARGET-PBG-TABLESPACE(Y | N)

When a source partition-by-growth table space has more partitions than the corresponding target, specify EXTEND-TARGET-PBG-TABLESPACE(Y) to add one or more partitions to target partition-by-growth table spaces. As many target partitions are added as needed to match the number of source partitions. If Y is specified, target partitions will be added even if DATA-MOVER(PGM(NONE)) is specified. Partitions are added using one of the following methods:

- For DB2 V9.1 subsystems, an UNLOAD and LOAD is used.
- For DB2 10 or later, an ALTER TABLE is used.

For table space cloning on DB2 V9.1 subsystems, the following UNLOAD and LOAD DD statements must be added to the source job. If the TCP/IP server job will be used, also add the following DD statements to the TCP/IP server job. Change the DD statements as required.

```
//CKZREC DD DSN=&h1q.SYSREC,
//          DISP=(NEW,CATLG,CATLG),
//          UNIT=SYSDA,SPACE=(CYL,(50,100),RLSE)
//CKZPUNCH DD DSN=&h1q.SYSPUNCH,
//          DISP=(NEW,CATLG,CATLG),
//          UNIT=SYSDA,SPACE=(CYL,(1,1))
//CKZSORT DD SPACE=(CYL,(50,100),RLSE),UNIT=SYSDA
//CKZUT1 DD SPACE=(CYL,(50,100),RLSE),UNIT=SYSDA
```

Note: CKZREC and CKZPUNCH may be preallocated.

Partitions are not added if the target partition-by-growth table space was created with the DEFINE NO clause and its data sets were not allocated. If an error occurs because the number of target partitions is greater than the number of source partitions, you should delete the extra target partitions.

- Default: N

- Required: No
- Restrictions: None
- Short form: XTPBG

FUZZY-COPY(Y | N)

The FUZZY-COPY parameter disallows DB2 Cloning Tool Table Space Cloning from stopping the source table spaces and index spaces before replicating them.

Refer to “How DB2 Cloning Tool Table Space Cloning starts and stops table spaces and index spaces” on page 156 for more information about stopping and starting table spaces and index spaces.

If Y, ADRDSSU is invoked by DB2 Cloning Tool Table Space Cloning with TOLERATE(ENQFAILURE) or the EMC API is invoked with TOLERATEENQFAILURE(Y). If you specify Y, note that RACF FACILITY class authority to process data sets even though shared or exclusive access fails.

Attention: This procedure may cause data integrity issues. Refer to “Considerations for using FUZZY-COPY” on page 160 for additional information about using this parameter.

- Default: N
- Required: No
- Restrictions: Valid only if DATA-MOVER(PGM(ADRDSSU) is specified (FlashCopy or SnapShot).
- Short form: FUZZY

IDCAMS-DDN(*ddname*)

The IDCAMS-DDN parameter specifies the name of a data set where DB2 Cloning Tool Table Space Cloning will write out IDCAMS commands to delete an object data set and rename the copied F0001 data set to the object data set name. The data set pointed to by IDCAMS-DDN must have an LRECL of 80 and RECFM of FB.

The purpose of this data set is to assist in replicating the DB2 VSAM objects when the DB2 table spaces and index spaces do not already exist on the target, either because the target object does not exist or was created with the DEFINE NO attribute.

- Default: None
- Required: No
- Restrictions: Valid when the table spaces or index spaces do not exist in the target subsystem.
- Short form: IDC-DDN

IGNORE-RF-MISMATCH-IF-NO-VAR-COLS(Y | N)

This parameter can be used to allow copy of table spaces from source to target when there is a mismatch involving reordered row format. For example, the source may have been migrated from DB2 V8 and be in basic row format (BRF), and the target objects may have been created on DB2 V9 NFM and be in reordered row format (RRF). When IGNORE-RF-MISMATCH-IF-NO-VAR-COLS is set to Y, table spaces with no variable columns can ignore the mismatch in row format and be copied without a warning. Variable columns are VARCHAR, LONGVAR, VARG, LONGVARG, and VARBIN.

When IGNORE-RF-MISMATCH-IF-NO-VAR-COLS is N, all reordered row mismatches are treated as mismatches. A copy of mismatched spaces is only allowed when ALLOW-COPY-ON-MISMATCH(Y). If ALLOW-COPY-ON-MISMATCH(Y) and IGNORE-RF-MISMATCH-IF-NO-VAR-COLS(Y), and there are one or more variable columns, a warning is issued and the space is copied. Note that any other incompatibility will override the processing of this command and ALLOW-COPY-ON-MISMATCH will determine if the space is copied.

- Default: None
- Required: No
- Restrictions: None
- Short form: VARCL

INCLUDE-ALL-RI (Y | N)

If set to Y, specifies that RI is to be added to all LISTDEFS. The RI indicator is automatically inserted in all LISTDEF statements.

- Default: N. If DDL-ENABLE(Y) and PROCESS-TYPE is Y, A, or G, the default is Y. If DATA-MASKING(Y), the default is Y.
- Required: N
- Restrictions: None
- Short form: INCRI

JOB-TEMPLATE (*inddname1, outddname1,...[, inddnamen, outddnamen]*)

The JOB-TEMPLATE parameter passes in DD name pairs of job templates. Up to 100 DD name pairs can be specified. These DD name pairs are processed independently of the PGM value (ADDRSSU, EMCAPL, or NONE) specified in the DATA-MOVER COPY subcommand. Non-SMS volume specification must be done manually when using job templates. See the sample member CKZJOB1 in the JCL library.

- Default: None.
- Required: N
- Restrictions: None
- Short form: JTEMP

LOG-APPLY (LA-ENABLE(Y | N)

[,COMMON-CONSISTENT-POINT(Y | N)]
 [,DATA-SHARING-MEMBERS(*id* | *ssid* | *zparm*)]
 [,END-POINT(TO_CURRENT | TO_QUIESCE | TO_LOGPOINT
X'byte_string')]
 [,MINILOG-HLQ(*miniloghighlevelqualifier*)]
 [,MINILOG-LARGE-FILE-TYPE(Y | N)]
 [,MINILOG-UNIT-TYPE(SYSALLDA)]
 [,MINILOG-QUANTITY-IN-TRACKS(Y | N)]
 [,MINILOG-PRIMARY-QUANTITY(*nnnn*)]
 [,MINILOG-SECONDARY-QUANTITY(*nnnn*)]
 [,MINILOG-VOLUME-COUNT(*nnn*)]
 [,MINILOG-DATACLAS(*dataclas*)]
 [,MINILOG-STORCLAS(*storclas*)]
 [,MINILOG-MGMTCLAS(*mgmtclas*)]
 [,NUMBER-OF-BUFFERS(*nn*)]
 [,NUMBER-OF-CHANNEL-PROGRAMS(*nn*)]
 [,QUIESCE-POINT(Y | N)]
 [,REBUILD-COPY-NO-INDEXES (Y | N)]
 [,SORTFILE-LARGE-FILE-TYPE(Y | N)]

```

| [ ,SORTFILE-UNIT-TYPE( SYSALLDA )]
| [ ,SORTFILE-PRIMARY-QUANTITY( nnnn )]
| [ ,SORTFILE-SECONDARY-QUANTITY( nnnn )]
| [ ,SORTFILE-VOLUME-COUNT( nnn )]
| [ ,SORTFILE-DATACLAS( dataclas )]
| [ ,SORTFILE-STORCLAS( storclas )]
| [ ,SORTFILE-MGMTCLAS( mgmtclas )]
| [ ,SORT-PROGRAM( DFSORT | SYNCSORT )]
| [ ,SPACES-PER-MINILOG( numberofspaces )]
| [ ,UNIFIED-WARNING( Y | N )]
| [ ,USE-TCPIP( Y | N )]
| [ ,WARN-IF-SKIP-QUIESCE( Y | N )]
| [ ,WARN-IF-TS-DEFINED-LOG-NO( Y | N )]
| [ ,WORKFILE-LARGE-FILE-TYPE( Y | N )]
| [ ,WORKFILE-UNIT-TYPE( SYSALLDA )]
| [ ,WORKFILE-QUANTITY-IN-TRACKS( Y | N )]
| [ ,WORKFILE-PRIMARY-QUANTITY( nnnn )]
| [ ,WORKFILE-SECONDARY-QUANTITY( nnnn )]
| [ ,WORKFILE-VOLUME-COUNT( nnn )]
| [ ,WORKFILE-DATACLAS( dataclas )]
| [ ,WORKFILE-STORCLAS( storclas )]
| [ ,WORKFILE-MGMTCLAS( mgmtclas )]
| [ ,ZPARAM-MEMBER( zparammembername )] )

```

LOG-APPLY is used in the source job and allows log records written by DB2 from before the copies in the source job until the target job is run to be applied to DB2 pages being updated in the target job.

LOG-APPLY is required when image copies are used as input to the cloning process; it is used to apply logs to the image copies and bring the cloned objects to the desired consistent point in time. Additional parameters can be used to specify the consistent point of the cloning process.

- Default: None
- Required: Yes, if PGM (SRCIMCPY) is specified.
- Restrictions: For cloning using PGM(ADRDSSU) or (EMCAPI), logs cannot be applied for LOBs, XML spaces, and index spaces. For cloning using PGM(SRCIMCPY), logs cannot be applied for XML spaces.
- Short form: LOGAP

LA-ENABLE- Set this parameter to Y to enable log apply. If this parameter is set to Y, you must include the ZPARAM-MEMBER parameter. If LA-ENABLE is Y and the source objects are in a data sharing group, you must specify the DATA-SHARING-MEMBERS parameter. If LA-ENABLE is N, the parameters are validated but no log changes will be applied to the target.

- Default: N
- Required: Yes, if PGM(SRCIMCPY) is specified.
- Restrictions: None
- Short form: LAENA

COMMON-CONSISTENT-POINT - Set this parameter to Y to have all table space objects brought to the same consistent point. For PGM(SRCIMCPY), if no common consistent point can be found, the spaces are not processed (unless UNIFIED-WARNING (Y) is specified).

- Default: N

- Required: No
- Restrictions: None
- Short form: COMCP

DATA-SHARING-MEMBERS - When the source subsystem is a data sharing group, identify the members of the group using this parameter. Specify an identification number, subsystem ID, and ZPARM member for each member in the data-sharing group. For example:

```
DATA-SHARING-MEMBERS (
  1,SS1A,SS1APARM
  2,SS1B,SS1BPARM )
```

- Default: None
- Required: Required if the source subsystem is a data sharing group
- Restrictions: None
- Short form: DSMBR

END-POINT - When PGM(SRCIMCPY), this parameter specifies when DB2 Cloning Tool Table Space Cloning is to stop looking for a consistent set of objects. This parameter is required for PGM(SRCIMCPY) and is ignored when using target job log apply. Specify one of the following:

- TO_CURRENT: (Default) Current point in time.
- TO_QUIESCE: Last (most recent) quiesce point.
- TO_LOGPOINT X'*byte-string*': The specified log point as 12 or 20 hexadecimal digits. If DSNJCNVT is set and 10-byte RBAs are in use, all byte strings must be 10 bytes. When this value is specified, logs are applied up to this specific log point.
- Default: ENDPOINT(TO_CURRENT)
- Required: No
- Restrictions: None
- Short form: ENDPT

MINILOG-HLQ - Specify the high level qualifier for the minilog data sets. Up to 35 characters are permitted; the last qualifier is automatically generated.

- Default: CKZ.MINILOG
- Required: No
- Restrictions: None.
- Short form: MLHLQ

MINILOG-LARGE-FILE-TYPE - Specify Y to indicate that dynamic allocation of the minilog data set should include the LARGE attribute. This allows for data sets to exceed 65,535 tracks.

- Default: N
- Required: No
- Restrictions: None
- Short form: MLTYPE

MINILOG-UNIT-TYPE - Specify the unit type for the minilog data set.

- Default: SYSALLDA
- Required: No
- Restrictions: None
- Short form: MLUNIT

MINILOG-QUANTITY-IN-TRACKS - Specify Y if the minilog is to be allocated in tracks or N if the minilog is to be allocated in cylinders.

- Default: N
- Required: No
- Restrictions: None
- Short form: MLTRK

MINILOG-PRIMARY-QUANTITY - Specify the minilog data set's primary quantity.

- Default: 250
- Required: No
- Restrictions: None
- Short form: MLPQT

MINILOG-SECONDARY-QUANTITY - Specify the minilog data set's secondary quantity.

- Default: 250
- Required: No
- Restrictions: None
- Short form: MLSQT

MINILOG-VOLUME-COUNT - Specify the maximum number of volumes that the minilog data set will require. If SMS parameters are specified to allow for multi-volume DASD data sets, this parameter must be at least as large as the number of volumes that the minilog will ultimately occupy. Enter a value between 1-255 inclusive, or blank to omit the volume count parameter.

- Default: blank
- Required: No
- Restrictions: None
- Short form: MLVOL

MINILOG-DATACLAS - If the minilog data set will be managed by SMS, specify the SMS Data Class.

- Default: None
- Required: No
- Restrictions: None
- Short form: MLDATA

MINILOG-STORCLAS - If the minilog data set will be managed by SMS, specify the SMS Storage Class.

- Default: None
- Required: No
- Restrictions: None
- Short form: MLSTOR

MINILOG-MGMTCLAS - If the minilog data set will be managed by SMS, specify the SMS Management Class.

- Default: None
- Required: No
- Restrictions: None
- Short form: MLMGMT

| NUMBER-OF-BUFFERS - If you want to adjust the BUFNO parameter in
| JCL, you can enter this parameter as a QSAM override.

- Default: 5
- Required: No
- Restrictions: None
- Short form: NRBUF

| NUMBER-OF-CHANNEL-PROGRAMS - If you want to adjust the NCP
| parameter (the number of read or write macro instructions) in JCL, use this
| parameter as a QSAM override.

- Default: 1
- Required: No
- Restrictions: None
- Short form: NRCHP

| QUIESCE-POINT - If this parameter is set to Y, a QUIESCE is issued after
| copies are complete.

- Default: N
- Required: No
- Restrictions: None
- Short form: QUIES

| REBUILD-COPY-NO-INDEXES - This parameter is used only for
| PGM(SRCIMCPY); it is ignored when using target job log apply. Set this
| parameter to Y to have indexes that were defined as COPY NO rebuilt in
| the target job.

- Default: N
- Required: No
- Restrictions: None
- Short form: RBCXX

| SORTFILE-LARGE-FILE-TYPE - Specify Y to indicate that dynamic
| allocation of the sort file data set should include the LARGE attribute. This
| allows for data sets to exceed 65,535 tracks.

- Default: N
- Required: No
- Restrictions: None
- Short form: SFTYPE

| SORTFILE-UNIT-TYPE - Specify the unit type for the sort file data set.

- Default: SYSALLDA
- Required: No
- Restrictions: None
- Short form: SFUNIT

| SORTFILE-QUANTITY-IN-TRACKS - Specify Y if the sort file is to be
| allocated in tracks or N if the sort file is to be allocated in cylinders.

- Default: N
- Required: No
- Restrictions: None
- Short form: SFTRK

| SORTFILE-PRIMARY-QUANTITY - Specify the sort file data set's primary
| quantity.

- Default: 250
- Required: No
- Restrictions: None
- Short form: SFPQT

| SORTFILE-SECONDARY-QUANTITY - Specify the sort file data set's
| secondary quantity.

- Default: 250
- Required: No
- Restrictions: None
- Short form: SFSQT

| SORTFILE-VOLUME-COUNT - Specify the maximum number of volumes
| that the sort file data set will require. If SMS parameters are specified to
| allow for multi-volume DASD data sets, this parameter must be at least as
| large as the number of volumes that the sort file will ultimately occupy.
| Enter a value between 1-255 inclusive, or blank to omit the volume count
| parameter.

- Default: blank
- Required: No
- Restrictions: None
- Short form: SFVOL

| SORTFILE-DATACLAS - If the sort file data set will be managed by SMS,
| specify the SMS Data Class.

- Default: None
- Required: No
- Restrictions: None
- Short form: SFDATA

| SORTFILE-STORCLAS - If the sort file data set will be managed by SMS,
| specify the SMS Storage Class.

- Default: None
- Required: No
- Restrictions: None
- Short form: SFSTOR

| SORTFILE-MGMTCLAS - If the sort file data set will be managed by SMS,
| specify the SMS Management Class.

- Default: None
- Required: No
- Restrictions: None
- Short form: SFMGMT

| SORT-PROGRAM - Specify the sort program to be used. Valid values are
| DFSORT (the default) or SYNCSORT.

- Default: DFSORT
- Required: No
- Restrictions: None
- Short form: SORTP

| SPACES-PER-MINILOG - Specify the number of spaces included in each
| minilog data set created by log apply. Only 36 minilog data sets can be
| created. If a value for SPACES-PER-MINILOG is small enough that more
| than 36 minilog data sets would be created, this parameter value will be
| raised. For example, if 1,000 base table spaces are to be processed and
| SPACES-PER-MINILOG is set to 25, 40 minilog data sets would be
| required. DB2 Cloning Tool Table Space Cloning will raise the
| SPACES-PER-MINILOG value to allow 36 or less minilog data sets to be
| create. In this example, 25 would be changed to 28, thereby creating 36
| minilog data sets.

- Default: 010
- Required: No
- Restrictions: None
- Short form: SPPML

| UNIFIED-WARNING - Set this parameter to Y to process commands even
| if some objects cannot be copied to the specified end point. If this
| parameter is set to Y with PGM(SRCIMCPY), and COMMON-
| CONSISTENT-POINT is set to Y, processing continues with a return code
| of 4 and warning messages are generated. If this parameter is set to N and
| no common consistent point is found, processing is halted and error
| messages are generated.

- Default: N
- Required: No
- Restrictions: None.
- Short form: UNWRN

| USE-TCPIP(Y|N) - This parameter specifies whether or not use TCP/IP to
| communicate with the source DB2 subsystem when cloning between DB2s
| on two different LPARs This parameter is required for cross LPAR log
| apply.

- Default: N
- Required: No
- Restrictions: Specify in the SOURCE job
- Short form: USETCP

| WARN-IF-SKIP-QUIESCE - If this parameter is set to Y, a warning message
| is issued if QUIESCE must be skipped for a status that prevents a
| QUIESCE point. The following states prevent a QUIESCE from being
| started for the table space(s) being copied: auxiliary CHECK-pending
| (ACHKP), CHECK-pending (CHKP), COPY-pending (COPY),
| REBUILD-pending (RBDP), and RECOVER-pending (RECP).

- Default: N
- Required: No
- Restrictions: None
- Short form: WIFSQ

| WARN-IF-TS-DEFINED-LOG-NO - This parameter is used only for
| PGM(SRCIMCPY); it is ignored when using target job log apply. When a
| base or LOB table space has the NOT LOGGED attribute, DB2 does not
| create logs for the space. This could result in errors when the source object
| is copied to the target using PGM(SRCIMCOPY). If WARN-IF-TS-
| DEFINED-LOG-NO is set to Y, a warning message is output for each table

| space with the NOT LOGGED attribute. If WARN-IF-TS-DEFINED-LOG-
| NO is set to N, an informational message is output for each table space
| with the NOT LOGGED attribute.

- Default: N
- Required: No
- Restrictions: None
- Short form: WXLOG

| WORKFILE-LARGE-FILE-TYPE - Specify Y to indicate that dynamic
| allocation of the work file data set should include the LARGE attribute.
| This allows for data sets to exceed 65,535 tracks.

- Default: N
- Required: No
- Restrictions: None
- Short form: WFTYPE

| WORKFILE-UNIT-TYPE - Specify the unit type for the work file data set.

- Default: SYSALLDA
- Required: No
- Restrictions: None
- Short form: WFUNIT

| WORKFILE-QUANTITY-IN-TRACKS - Specify Y if the work file is to be
| allocated in tracks or N if the work file is to be allocated in cylinders.

- Default: N
- Required: No
- Restrictions: None
- Short form: WFTRK

| WORKFILE-PRIMARY-QUANTITY - Specify the work file data set's
| primary quantity.

- Default: 250
- Required: No
- Restrictions: None
- Short form: WFPQT

| WORKFILE-SECONDARY-QUANTITY - Specify the work file data set's
| secondary quantity.

- Default: 250
- Required: No
- Restrictions: None
- Short form: WFSQT

| WORKFILE-VOLUME-COUNT - Specify the maximum number of volumes
| that the work file data set will require. If SMS parameters are specified to
| allow for multi-volume DASD data sets, this parameter must be at least as
| large as the number of volumes that the sort file will ultimately occupy.
| Enter a value between 1-255 inclusive, or blank to omit the volume count
| parameter.

- Default: blank
- Required: No
- Restrictions: None

- Short form: WFMVOL

WORKFILE-DATACLAS - If the work file data set will be managed by SMS, specify the SMS Data Class.

- Default: None
- Required: No
- Restrictions: None
- Short form: WFDATA

WORKFILE-STORCLAS - If the work file data set will be managed by SMS, specify the SMS Storage Class.

- Default: None
- Required: No
- Restrictions: None
- Short form: WFSTOR

WORKFILE-MGMTCLAS - If the work file data set will be managed by SMS, specify the SMS Management Class.

- Default: None
- Required: No
- Restrictions: None
- Short form: WFMGMT

ZPARAM-MEMBER - This parameter specifies the ZPARAM member name for the source subsystem. This parameter is required when LA-ENABLE is Y.

- Default: None
- Required: Required if LA-ENABLE =Y
- Restrictions: None
- Short form: ZPARAM

LONGVAR-COMPATIBILITY(Y | N)

This keyword prevents a mismatch message when running a source job in which the source objects have LONGVAR columns and the target objects have corresponding VARCHAR columns (or vice versa). This situation may occur if objects have been migrated to DB2 Version 9.1. After migration, existing LONGVAR columns remain as LONGVAR; however, new LONGVAR columns become VARCHAR columns. When these objects are contained in a source job, DB2 Cloning Tool Table Space Cloning will issue a mismatch warning. If you specify Y for LONGVAR-COMPATIBILITY, the mismatch message and its corresponding return code of 4 are suppressed.

Note: The lengths of the corresponding columns must be the same; if they are not, data may be truncated or a DB2 abend may occur.

- Default: N
- Required: No
- Restrictions: None
- Short form: LONGV

OBJECT-MISMATCH-RETURN-CODE(*mismatchctype1,returncode1, . . . mismatchctypen,returncoden*)

This keyword allows you to change the return code associated with a particular object mismatch. Mismatches are detected after the source and target objects are identified and before the copy starts.

Individual mismatches may be assigned a return code of 0, 4, or 8. If the return code is 0, an informational message is issued and the object is treated as if there is no mismatch. If the return code is 4, warning message is issued. If the return code is 8, an error message is issued and no copies are performed, regardless of the setting for ALLOW-COPY-ON-MISMATCH. The default return code is 4 if the mismatch return code command is not specified.

mismatchtype must be one of the following:

- TS_CC SID
- TS_DSSIZE
- TS_PARTITIONS
- TS_PG SIZE
- TS_NT ABLES
- TS_SEG SIZE
- TS_TY PE
- TS_ORG ANIZATIONTYPE
- TS_CL ONE
- TS_HASHSPACE
- TP_COMPRESS
- TP_FORMAT
- TB_COLCOUNT
- TB_EDPROC
- TB_HASHKEYCOLUMNS
- CL_COLNO
- CL_COLTYPE
- CL_LENGTH
- CL_SCALE
- CL_DEFAULT
- CL_FLDPROC
- CL_SOURCETYPEID
- CL_SEQTYPE
- CL_CYCLE
- CL_CACHE
- CL_START
- CL_MINVALUE
- CL_MAXVALUE
- CL_INCREMENT
- CL_HASHKEY_COLSEQ
- IX_UNIQUERULE
- IX_COLCOUNT
- IX_PG SIZE
- IX_INDEXTYPE
- IX_PIECESIZE
- IX_NR_PARTS
- IX_CLUSTERING
- IX_HASH

- IX_SPARSE
- IX_UNIQUE_COUNT
- IX_COLNO
- IX_COLSEQ
- IX_ORDERING
- IP_LIMITKEY

Note: If you specify IX_COLNO, IX_COLSEQ, and IX_ORDERING, you must also specify CHECK-INDEX-KEYS(YES).
returncoden must be 0, 4, or 8.

- Default: If the mismatch return code command is not specified, the default return code is 4.
- Required: No
- Restrictions: None
- Short form: OMMRC

OBJECT-TRANSLATE(

DATABASE, sourcedatabase, targetdatabase
 [*DATABASE, sourcedatabasen, targetdatabasen*]
TABLESPACE, sourcetablespace1, targettablespace1 . . .
 [*TABLESPACE, sourcetablespacen, targettablespacen*]
TABLE, sourcetable1, targettable1 . . .
 [*TABLE, sourcetablen, targettablen*]
INDEXSPACE, sourceindexspace1, targetindexspace1 . . .
 [*INDEXSPACE, sourceindexspacen, targetindexspacen*]
INDEX, sourceindex1, targetindex1 . . .
 [*INDEX, sourceindexn, targetindexn*]
CREATOR, sourcecreator1, targetcreator1 . . .
 [*CREATOR, sourcecreatorn, targetcreatorn*]
VCAT, sourcevcat1, targetvcat1 . . .
 [*VCAT, sourcevcatn, targetvcatn*])

The OBJECT-TRANSLATE parameter allows DB2 Cloning Tool Table Space Cloning to map the source objects to target objects with different names. This feature allows DB2 Cloning Tool Table Space Cloning to copy table spaces and index spaces to the same subsystem or to a different subsystem with different names.

The DATABASE, INDEXSPACE, TABLESPACE, INDEX and TABLE parameters specify the names of the source objects and the new target object names, in pairs.

The CREATOR parameter specifies the creator name of the source table spaces or index spaces and the new creator name for the target table spaces or index spaces.

The VCAT parameter specifies the VCAT name of the source table spaces or index spaces and the new VCAT name for the target table spaces or index spaces. Note that DEFVCAT in the TARGET-DB2 parameter is still used if a VCAT is unavailable after applying all OBJECT-TRANSLATE VCAT commands. VCAT mapping is not required when target objects exist; in that case, the source database name and space name are used to find target objects and their VCAT values.

DB2 Cloning Tool Table Space Cloning will verify the source names on the source DB2. When found, the supplied target names are used to verify compatible target table spaces and index spaces on the target DB2.

If the INDEXSPACE parameter is specified, DB2 Cloning Tool Table Space Cloning uses the Database.Indexspace name to find the target index. If INDEXSPACE is not specified, then the Creator.Name is used to find the target index.

The source and object pairs may be specified using masking. Specify masking as follows:

- Percent sign (%) or asterisk (*) represents n characters.
- Underscore (_) or question mark (?) represents a single character. Use the question mark (?) rather than the underscore (_) for creator, table and index names, as the underscore is a valid character for these three object names.

Refer to the topic “OBJECT-TRANSLATE considerations” on page 504 for additional information.

- Default: None
- Required: No
- Restrictions: None
- Short form: OBJXLATE

```
PROCESS-DDL( DDL-ENABLE( Y | N )
[ ,PROCESS-TYPE( [ Y | N | G | X | A] ) ]
[ ,PROCESS-DDL-DDN( ddname ) ]
[ ,IGNORE-CREATE-OBJECT-EXISTS( Y | N ) ]
[ ,GENERATE-DDL-DEFAULTS( Y | N ) ]
[ ,DDL-ATTRIBUTE-
CHANGE(attributename,srcvalue,trgvalue,applyTOTYPE,applyTOOBJECT) ] )
```

PROCESS-DDL generates and/or executes DDL to be used for creating missing target objects. CREATE DDL can be generated for databases, table spaces, tables, and indexes; LOB and XML spaces are supported. All referenced storage groups, distinct types and other supporting objects must exist on the target to be able to execute the generated DDL. In addition to supporting missing target objects, DB2 Cloning Tool Table Space Cloning can generate source object DDL to be saved to a data set or can execute DDL from an input data set.

For more information about the DDL generation process, refer to the topic “Considerations for generating target object DDL using PROCESS-DDL” on page 160.

- Default: N
- Required: No
- Restrictions: None
- Short form: DDL

DDL-ENABLE - Set this parameter to Y to enable DDL processing. If set to N, DDL processing is disabled; this allows you to leave the DDL commands in the source JCL without processing them.

- Default: N
- Required: No
- Restrictions: None
- Short form: DDLENA

PROCESS-TYPE - Specify the processing type.

- Y: Generate and execute DDL for non-existing target objects.
- N: Do not generate or execute DDL. This is the default.

- G: Generate DDL for missing target objects, but do not execute.
- X: Do not generate DDL, but execute DDL already in PROCESS-DDL-DDN(*ddname*).
- A: Generate all source object DDL, but do not execute.

If the processing type generates DDL (Y, G and A), generated DDL is written into PROCESS-DDL-DDN(*ddname*), if it exists. For the A and G commands, the PROCESS-DDL-DDN is required.

To prevent copies to the target subsystem using the F0001 data sets, run PROCESS-TYPE(G) and PROCESS-TYPE (A) with PGM(NONE). Once the target DDL is correct, then submit the copy jobs.

If the processing type is Y or G, and multiple VCATs are required to map data set names from source to target, you must specify the OBJECT-TRANSLATE VCAT parameter.

- Default: N
- Required: No
- Restrictions: None
- Short form: PTYPE

PROCESS-DDL-DDN - Specify the DD name where the generated DDL will be written to or read from.

- Default: None
- Required: Yes, unless PROCESS-TYPE is Y or N
- Restrictions: None
- Short form: DDL-DDN

IGNORE-CREATE-OBJECT-EXISTS - This parameter can be used when PROCESS-TYPE is X or Y. If IGNORE-CREATE-OBJECT-EXISTS is set to Y, -601 SQL errors are ignored.

- Default: N
- Required: No
- Restrictions: None
- Short form: IGOBJ

GENERATE-DDL-DEFAULTS - When set to Y, this parameter generates DDL statements for default values. If set to N, the statements are omitted. For example, if GENERATE-DDL-DEFAULTS is set to N and the catalog value for CLOSE is Y (the default), CLOSE YES is not included in the DDL stream. If GENERATE-DDL-DEFAULTS is set to Y, CLOSE YES is included in the DDL stream.

- Default: N
- Required: No
- Restrictions: None
- Short form: GDDL

DDL-ATTRIBUTE-CHANGE - This parameter allows values not changed with object translate to be changed from source to target. Refer to the topic "DDL-ATTRIBUTE-CHANGE parameter values" on page 506 for a detailed explanation of this parameter.

- Default: None
- Required: No
- Restrictions: None

- Short form: DDLAC

REPLACE-TARGET-DSN(Y | N)

The REPLACE-TARGET-DSN parameter over-writes the target VSAM object if it exists and “Y” is specified.

Note: For this parameter, the I and J data sets are considered the same data set. For example, if the data set DSN091D.DSNDBC.DAHDB.DAH2TS.J0001.A001 exists on the target and DSN091D.DSNDBC.DAHDB.DAH2TS.I0001.A001 is the target data set name, the copy will not be allowed if REPLACE-TARGET-DSN(N) is specified.

- Default: Directed by CKZINI token REPLACE_TARGET_DSN in :DSN_COPY_OPTIONS section. If no token value, then Y.
- Required: No
- Restrictions: None
- Short form: RTD

RESET-LOGRBA(Y | N)

The RESET-LOGRBA parameter is passed to the DB2 Cloning Tool Table Space Cloning target job and allows DB2 Cloning Tool Table Space Cloning to reset the LOGRBA. The LOGRBA will always be reset if there are OBID changes to be made. The level IDs in the target VSAM objects are always reset to prevent DB2 down-level rejection of the target VSAM objects.

- Default: Directed by CKZINI token RESET_LOGRBA in :DSN_COPY_OPTIONS section. If no token value, then Y.
- Required: No
- Restrictions: None
- Short form: RL

SIMULATE(A | N | Y)

The SIMULATE parameter allows DB2 Cloning Tool Table Space Cloning to be run in one of three modes.

A - if PGM(ADRDSSU) or PGM(EMCAPI), stop target and then source spaces, call the data mover program in NORUN mode, start source and target spaces and write out SYNCDB2 commands for the target; if PGM(NONE), validate target table spaces and index spaces and write out SYNCDB2 commands for the target.

N - normal mode

Y – (the default) process LISTDEF to get source table spaces and index spaces and stop.

Refer to “How DB2 Cloning Tool Table Space Cloning starts and stops table spaces and index spaces” on page 156 for more information about stopping and starting table spaces and index spaces.

- Default: Y
- Required: No
- Restrictions: SIM(A) is not valid with PGM(SRCIMCPY).
- Short form: SIM

CAUTION:

DDL processing cannot be simulated. If you do not want DDL to be processed during a simulation, change DDL-ENABLE to N.

SQLOUT-DDN(*ddname*)

The SQLOUT-DDN parameter supplies the *ddname* assumed via JCL to point at the data set which will be passed to the DB2 Cloning Tool Table Space Cloning target job to update the target DB2 catalog with identity column sequence information. The data set pointed to by SQLOUT-DDN must have an LRECL of 80 and RECFM of FB.

- Default: None
- Required: Yes, if one or more tables has an identity column with a different start value requiring an ALTER TABLE to be issued in the target job.
- Restrictions: None
- Short form: SQL-DDN

START-SOURCE-DDN(*ddname*)

The START-SOURCE-DDN parameter specifies the name of a data set where DB2 Cloning Tool Table Space Cloning will write out DB2 START commands for all the source table spaces and index spaces. The data set pointed to by START-SOURCE-DDN must have an LRECL of 80 and RECFM of FB.

The purpose of this data set is to assist in copying the VSAM objects outside of DB2 Cloning Tool Table Space Cloning.

- Default: None
- Required: No
- Restrictions: Valid only if DATA-MOVER(PGM(NONE)) is specified.
- Short form: STARS-DDN

STOP-SOURCE-DDN(*ddname*)

The STOP-SOURCE-DDN parameter specifies the name of a data set where DB2 Cloning Tool Table Space Cloning will write out DB2 STOP commands for all the source table spaces and index spaces. The data set pointed to by STOP-SOURCE-DDN must have an LRECL of 80 and RECFM of FB.

The purpose of this data set is to assist in copying the VSAM objects outside of DB2 Cloning Tool Table Space Cloning.

- Default: None
- Required: No
- Restrictions: Valid only if DATA-MOVER(PGM(NONE)) is specified.
- Short form: STOPS-DDN

STOP-TARGET-DDN(*ddname*)

The STOP-TARGET-DDN parameter specifies the name of a data set where DB2 Cloning Tool Table Space Cloning will write out DB2 STOP commands for all the target table spaces and index spaces. The data set pointed to by STOP-TARGET-DDN must have an LRECL of 80 and RECFM of FB.

The purpose of this data set is to assist in copying the VSAM objects outside of DB2 Cloning Tool Table Space Cloning.

- Default: None
- Required: No
- Restrictions: Valid only if DATA-MOVER(PGM(NONE)) is specified
- Short form: STOPT-DDN

SYNCDB2-DDN(*ddname*)

The SYNCDB2-DDN parameter supplies the *ddname* assumed via JCL to point at the data set which will be passed to the DB2 Cloning Tool Table Space Cloning target job for ID translation and to make the VSAM objects accessible to the target DB2. This parameter is required if the target job must be run to reset LOG RBAs or change object IDs or both. The data set pointed to by SYNCDB2-DDN must have an LRECL of 80 and RECFM of FB.

- Default: None
- Required: No. This is the input to the target job. Required only if any changes, such as data masking, new OBIDs or resetting the LOGRBA, are to be made to the target data sets after being copied.
- Restrictions: None
- Short form: SYN-DDN

TARGET-JOB-INDEX-REBUILD-DDN(*ddname*)

TARGET-JOB-INDEX-REBUILD-DDN can be used to rebuild all indexes whose tables were affected by data masking or log apply page changes. Any table in the target job that has a page changed via data masking or log apply requires index(es) to be rebuilt. TARGET-JOB-INDEX-REBUILD-DDN specifies the input and output DDs to be used to generate REBUILD INDEX utility jobs.

To use this functionality, you must modify a sample template provided by DB2 Cloning Tool Table Space Cloning and add DDs to the target job. For detailed instructions on using this keyword to rebuild indexes, refer to the topics Chapter 17, "Using data masking with table space cloning," on page 239, Chapter 19, "Using LOG-APPLY to make consistent copies of table spaces and index spaces," on page 265, and "Options for rebuilding indexes when cloning table spaces and index spaces from image copies" on page 262. When the target job completes, submit the output to rebuild all affected indexes.

- Default: None
- Required: No
- Restrictions: None
- Short form: TRGIR

TARGET-JOB-REPAIR-DDN(*ddname*)

TARGET-JOB-REPAIR-DDN is used to generate a REPAIR job to correct differences between the DB2 catalog and one or more page sets, and to specify whether DB2 Cloning Tool Table Space Cloning or the user submits the job. Specify one to seven characters of the target job DD name; the DDNAME is suffixed with .I and .O when added to the template.

- Default: None
- Required: No
- Restrictions: None
- Short form: TRGJR

V7-MIGRATED-OBJECTS-PRESENT (*Y | N*)

The V7-MIGRATED-OBJECTS-PRESENT parameter is required if DB2 Cloning Tool Table Space Cloning will be copying migrated DB2 V7 objects on a DB2 Version 8 or DB2 Version 9.1 subsystem.

Note: Setting this keyword to Y will cause source job performance degradation during the discovery phase.

- Default: N
- Required: No
- Restrictions: None
- Short form: V7MOP

WARN-IF-OBJECT-NOT-TRANSLATED (Y | N)

The WARN-IF-OBJECT-NOT-TRANSLATED parameter specifies that DB2 Cloning Tool Table Space Cloning is to check that each source object is translated to a new target value. Source objects that are not translated result in a warning message. Use this with PGM(NONE) to check that all masks are set correctly.

- Default: N
- Required: No
- Restrictions: None
- Short form: WIOXT

WARN-ON-DATASET-EXTENSION-MISMATCH(Y | N)

The WARN-ON-DATASET-EXTENSION-MISMATCH parameter provides the ability to specify the type of message that is issued when the number of data set extensions differs from source to target. If you specify WARN-ON-DATASET-EXTENSION-MISMATCH(Y), a warning message is issued and the job return code is set to 4. If you specify N, an informational message is issued.

- Default: N
- Required: No
- Restrictions: None
- Short form: WODXM

WARN-ON-INCOMPLETE-RI (Y | N)

The WARN-ON-INCOMPLETE-RI parameter specifies that when one or more LISTDEF statements do not specify RI, a warning message is issued.

- Default: N
- Required: No
- Restrictions: None
- Short form: WARNONICRI

WARN-ON-SIMPLE-TABLESPACE (Y | N)

This parameter allows you to be warned when simple table spaces are to be copied. When N is specified, an informational message is printed for each simple table space found on the source or target subsystem. When Y is specified, a warning message is issued.

In some cases when a simple table space is copied by DB2 Cloning Tool Table Space Cloning, the copy may have duplicate rows. This is caused by incompatibilities between table spaces migrated from an earlier DB2 version and processed using DB2 Version 9.1 or later.

When a simple table space is copied for the first time, ensure the target table space is accessible and has the correct number of rows. Once all of the simple table spaces in a job are verified, use the default N to eliminate warning messages.

- Default: N
- Required: No
- Restrictions: None

- Short form: WOSTS

WARN-ON-VERSIONS (Y | N)

Use this parameter to request an informational or warning message for all table spaces and index spaces that have a non-zero version count. These messages may indicate that a REORG and REPAIR utility must be run along with DB2 Cloning Tool Table Space Cloning processing. When set to Y and a table space or index space is found to have a non-zero version number, a warning message is output. When set to N, an informational message is output.

- Default: N
- Required: No
- Restrictions: None
- Short form: WVERS

XMLSTRING-DDN (*ddname*)

The XMLSTRING-DDN parameter passes the source subsystem XMLSTRINGS catalog table contents from the source job to the target job. The data set pointed to by XMLSTRING-DDN must have an LRECL of 80 and RECFM of FB.

- Default: None
- Required: Required if the source subsystem tables are being copied to a different subsystem, and one or more XML columns are present in the source subsystem tables.
- Restrictions: None
- Short form: XML-DDN

OBJECT-TRANSLATE considerations

This topic describes specifics about using the OBJECT-TRANSLATE keyword.

Mask syntax

The source and object pairs in the OBJECT-TRANSLATE keyword may specify masking. The table that follows shows the allowable filter characters.

Table 60. Filter characters allowed for source and object pairs filter masks

Character	Description
* or %	A single asterisk or percent sign represents 0 to nn characters of any value.
_ or ?	An underscore or question mark represents one non-blank character. Use the question mark (?) rather than the underscore (_) for creator, table, and index names, as the underscore is a valid character for these object names.

When using masks, the order of the object translate specifications for each object type becomes critical. Masking must be in the order of most restrictive to least restrictive. If not, the wrong target name could be selected for the corresponding source name.

Other masking rules are:

- The order, number, and type of mask characters must be the same between the source and target object

- `%%`, `%_`, or `_%` are invalid masks if neither the source nor target has a non-masking character in between. However, a double underscore (`__`) is a valid mask.

The following table shows examples of valid and invalid masks for the `OBJECT-TRANSLATE` keyword.

Table 61. Masks for the `OBJECT-TRANSLATE` keyword

Source Mask	Target Mask	Valid or Invalid	Comments
<code>%aa_a%</code>	<code>_aaa_a%</code>	Invalid	Source begins with <code>%</code> , target begins with <code>_</code>
<code>_aa_a%</code>	<code>_aaa_a</code>	Invalid	Source ends with <code>%</code> , target does not end with <code>%</code>
<code>_aa_%</code>	<code>_aaa_%</code>	Invalid	Neither source nor target has a character between <code>_</code> and <code>%</code>
<code>_aa%%%</code>	<code>_aaa%%%</code>	Invalid	Neither source nor target has a character between <code>%%%</code>
<code>_aa_%</code>	<code>_aaa_a%</code>	Valid	
<code>_aa_a%</code>	<code>_aaa_%</code>	Valid	
<code>_aa%_</code>	<code>_aaa%a_</code>	Valid	
<code>_aa%a%</code>	<code>_aaa%%%</code>	Valid	
<code>_aa_%</code>	<code>_aaa_a%</code>	Valid	
<code>aa__a</code>	<code>aa_a__a</code>	Valid	
<code>aa_a__a</code>	<code>aa__a</code>	Valid	

Syntax example

```
OBJECT-TRANSLATE (
CREATOR,%00?_,%0N?_,
TABLESPACE,Z@50_T_,Z#50_U_,
DATABASE,Z@50_DB,Z@50_EB,
DATABASE,Z@50_DB_,Z@50_DB_,
DATABASE,Z#50%B_,Z#50%C_,
DATABASE,Z@50_D%,Z#50_D%,
CREATOR,%N?_,%N?_,
TABLE,%TB11,%TB12,
INDEX,Z%,Y%
VCAT,%P,%T
)
```

Specifying a database object translate mask and including indexspaces in LISTDEF

In some cases, when using object translation to change a database name, target indexspaces may not be found. This is because DB2-generated indexspace names may not necessarily be identical from the source to the target subsystems. For example, `INDEXA` on the source may not have the identical indexspace name as `INDEXA` on the target. The translated database cannot be used unless an indexspace object translation value for the indexspace is provided with the translated database. With masking available for object translate, if all the indexspace names are the same from the source to the target, specify:

```
OBJECT-TRANSLATE (database,a,b,indexspace,%,%)
```

and *database.indexspace* will be used to locate all the target indexspaces. If all the indexspace names are not the same across subsystems, then specify the indexspaces that are different before *%,%* in the object translate command; for example:

```
OBJECT-TRANSLATE  
(database,a,b,indexspace,abc,efg,indexspace,jkl,mno,indexspace,%,%)
```

DDL-ATTRIBUTE-CHANGE parameter values

The DDL-ATTRIBUTE-CHANGE parameter allows values not changed with object translate to be changed from source to target. While OBJECT-TRANSLATE commands allow the pairing of source to target objects for copying purposes, DDL-ATTRIBUTE-CHANGE parameters allow other attributes of the selected objects to be changed.

The syntax for the DDL-ATTRIBUTE CHANGE parameter is:

```
DDL-ATTRIBUTE-CHANGE (attributename,srcvalue,trgvalue,applytotype,applytoobject)
```

The possible parameter values are detailed in the following section.

attributename

This parameter value names the attribute that can be changed. Refer to Table 62 on page 507 for a list of possible values.

srcvalue

This parameter identifies the source value to be changes. It can be blank; if left blank, the change applies to all source values of that attribute.

trgvalue

This parameter identifies the value that the attribute will be in the target. It can be blank; if left blank, the source values are changed to the default for that attribute, if one exists.

Note: There is no default for BUFFERPOOL.

applytotype

This parameter specifies the type of object that the attribute change will be applied to. *applytotype* can be:

- DATABASE
- TABLESPACE
- TABLEPART
- TABLE
- INDEX
- INDEXPART

If not specified, *applytotype* defaults to all object types that the *attributename* applies to.

applytoobject

This parameter names the object(s) that this change applies to. *applytoobject* references the source object names. It defaults to % or all objects of the specified type. *applytoobject* can also be a mask. To specify all object names, a blank or % may be used. When using a mask, it must refer to one of the following objects: database, table space, table, or index. For masking, % represents zero or more characters and ? represents one character. When using TABLEPART or INDEXPART as the *applytotype*, use the TABLESPACE or INDEXSPACE name for *applytoobject*.

Note: Leaving both *applytotype* and *applytoobject* blank applies the change to all object types and names that the *attributetype* applies to. Therefore, instead of an underscore, a question mark must be used to represent a single mask character.

The following table lists the *attributename* parameters, their possible values (*srcvalue* and *trgvalue*), and the type of object that the parameter may apply to (*applytotype*):

Table 62. Parameter values by attributename

attributename	srcvalue/trgvalue	applytotype
STOGROUP	For DB2 Version 8 ENFM, Version 9.1 and later: 128 chars For DB2 Version 8 CM: 8 chars	Database, table space partition, index partition
GBPCACHE	CHANGED ALL SYSTEM NONE	Table space partition
	CHANGED ALL NONE	Index partition
LOG	YES NO	Table space
	Note: LOG YES or NO syntax is generated for DB2 Version 8 LOGGED YES or NO syntax is generated for DB2 Version 9.1 and later	
PRIQTY	1-7 decimal digits or -1	Table space partition, index partition
SECQTY	1-7 decimal digits or -1	Table space partition, index partition
	Note: PRIQTY and SECQTY are matched using the catalog value of PQTY and SECQTYI, not what is entered using the DDL. DDL generation multiplies this catalog value by 4 to get the PRIQTY and SECQTY DDL values. For user managed data sets, RUNSTATS must have been run to populate the fields in the catalog.	
TRACKMOD	YES NO	Table space partition
CLOSE	YES NO	Table space, index
DATA_CAPTURE	NONE CHANGES	Table
BUFFERPOOL	<i>pool name</i>	Database, table space, index

The following table provides a cross reference of object type (*applytotype*) to the attribute that can be changed (*attributename*).

Table 63. Possible attributename parameters by object type

applytotype	attributename
Database	STOGROUP, BUFFERPOOL
Table space	LOG, CLOSE, BUFFERPOOL
Table space partition	STOGROUP, GBPCACHE, PRIQTY, SECQTY, TRACKMOD
Table	DATA_CAPTURE
Index	CLOSE, BUFFERPOOL

Table 63. Possible attributename parameters by object type (continued)

applytotype	attributename
Index partition	STOGROUP, GBPCACHE, PRIQTY, SECQTY

Additional information about specifying DDL-ATTRIBUTE-CHANGE

DDL-ATTRIBUTE-CHANGE commands should be in order from the most restrictive to the least restrictive. Restrictions are specified using the *applytoobject* or mask. For example, assume the following two commands where all table space objects beginning with ZYX are included in the *applyto* mask, and that they be matched with the first LOG command:

```
DDL-ATTRIBUTE-CHANGE(           -
LOG,,YES, TABLESPACE,ZYX%,    -
LOG,,NO,,,                     -
)
```

The table spaces will all be changed to LOG YES or remain LOG YES. Therefore, the second LOG command will change all those table spaces not beginning with ZYX from LOG YES to LOG NO, or they will remain LOG NO. Note that in this example, including "TABLESPACE" as the *applytotype* is not required because LOG can only be specified for table spaces. Most other attributes pertain to multiple object types and should be explicitly specified. For example, to only change GBPCACHE for index spaces, the INDEXSPACE *applytotype* should be specified in the command.

PRIQTY and SECQTY are matched using the catalog value of PQTY and SECQTYI, not what is entered using DDL. DDL generation multiplies this catalog value by 4 to get the PRIQTY and SECQTY DDL values. For user managed data sets, RUNSTATS must have been run to populate the fields in the catalog.

Specification exceptions

- An individual partition cannot be referenced. All partitions are checked for any attribute changes. If, however only one partition has a particular GBPCACHE value, for example, and there is an applicable DDL-ATTRIBUTE-CHANGE GBPCACHE command, no other partition will be affected.
- If a table space has the LOG NO attribute, DATACAPTURE CHANGES cannot be specified.
- Space attributes cannot be changed for LOB or XML spaces.

If attribute changes are required that are not supported by DB2 Cloning Tool Table Space Cloning, in the source job specify DDL(G) to generate the target DDL and PGM(NONE). After this source job has run, edit the DDL and then submit on the target subsystem. Then rerun the source job to access the new target objects.

GLOBAL command values for EMC TimeFinder/Clone Mainframe Snap Facility data set level support

This topic discusses considerations for EMC's GLOBAL command values when using the TimeFinder/Clone Mainframe Snap Facility's data set level support to make the copies.

GLOBAL command parameters

When DB2 Cloning Tool Table Space Cloning initiates EMC copies using the EMC API, most GLOBAL command parameter values use the EMC default. Exceptions are noted in the table that follows.

Table 64. GLOBAL parameter defaults setting by DB2 Cloning Tool Table Space Cloning

GLOBAL parameter	DB2 Cloning Tool Table Space Cloning setting
NQWAIT	NO
NOTIFYWHENCOMPLETE	DATASET
REUSE	YES
TOLERATEALLOCATIONFAILURE	YES
TOLERATECOPYFAILURE	YES
TOLERATEREUSEFAILURE	YES
VERIFY	NO
VSAMENQMODE	NONE

The GLOBAL values listed in the following table are set during processing using the DB2 Cloning Tool Table Space Cloning parameters.

Table 65. GLOBAL values set during processing by DB2 Cloning Tool Table Space Cloning

EMC GLOBAL value	DB2 Cloning Tool Table Space Cloning setting
HOSTCOPYMODE	If FUZZY-COPY(N), this parameter value is set to EXCLUSIVE. Otherwise, it is set to SHARED
MAXRC	This parameter always uses the DB2 Cloning Tool Table Space Cloning MAX-COPY-RC value.
REPLACE	This parameter is set to YES if REPLACE-TARGET-DSN(Y). Otherwise, it is set to NO.
TOLERATEENQFAILURE	If FUZZY-COPY(N), this parameter is set to NO. Otherwise, it is set to YES.
TYRUN	If SIM(A), this value is set to NORUN. Otherwise, it is set to RUN.

If EMC GLOBAL values other than those previously described are required, then you may use JOB-TEMPLATES to create an EMC TimeFinder/Clone SNAP DATASET job and submit it after the source job has run. Unlike ADRDSSU, an EMC job template cannot be automatically submitted using the CMDDDNAME parameter of the COPY DATA-MOVER command. That is due to the differences between the EMC API and EMC TimeFinder/Clone data set snap support. Refer to the JCL sample CKZJOB3 for a sample of an EMC data set snap input template. This sample is similar to the EMC API job internally submitted by DB2 Cloning Tool Table Space Cloning. This template assumes the following DB2 Cloning Tool Table Space Cloning COPY parameters are in effect:

```
FUZZY-COPY(N)
MAX-COPY-RC(0)
REPLACE-TARGET-DSN(Y)
```

Other changes to the DB2 Cloning Tool Table Space Cloning COPY command that must be made are:

- Set COPY subcommand DSNS-PER-COPY to 1.
- Set COPY subcommand DSS-COPY-COMMANDS to the number of data sets you wish to copy in a single SNAP job step. The maximum is 256, which is the recommended setting.

Change your input template variable and submit the source job to create the output job template. After the DB2 Cloning Tool Table Space Cloning source job has successfully run and you have verified that the SNAP parameters are correct, submit the output template job to perform the copies using EMC SNAP DATASET.

DATASUBTYPE

The DATASUBTYPE command is optional.

DATASUBTYPE may be used to specify the data subtype value (BIT, MIXED, or SBCS) needed by data masking.

The DATASUBTYPE command is useful when you are specifying data masking for a column that has been migrated from an earlier version of DB2 and has a blank for FOREIGNKEY in SYSIBM.SYSCOLUMNS.

If a source column has data masking specified and either the source or target has a blank for the FOREIGNKEY column in SYSCOLUMNS, the non-blank value will be used by data masking. This will not be reported as an object mismatch.

If both source and target have different non-blank values, a mismatch will be reported unless the source value is overridden by a DATASUBTYPE command. The source value for FOREIGNKEY will be used to perform data masking.

If the DATASUBTYPE command is found for a column, it overrides whatever FOREIGNKEY values exist in the catalog. If the source and target catalogs have the same non-blank value, but it differs from the DATASUBTYPE command value, the DATASUBTYPE command value still is used. No warning message will be issued.

If no FOREIGNKEY value is specified in the catalog or by command, SBCS will be used.

If the column specified in the masking definition is not a string column, any DATASUBTYPE command for this table and column is ignored. String columns are defined with TYPEDEF values CHAR, VARCHAR, LONGVAR, CLOB, DBCLOB and DISTINCT where the DATATYPEID is a string type.

DATASUBTYPE command syntax

DATASUBTYPE

Required keywords:

```
TABLECREATOR( tblcreator )  
TABLENAME( tblname )  
COLUMNNAME( colname )  
SUBTYPE( B | M | S )
```

DATASUBTYPE command & keyword definitions

Required and conditionally required keywords are described first, followed by optional keywords

DATASUBTYPE

The DATASUBTYPE command is optional. This command may be used to specify the data subtype value. This value may be needed by data masking when a column has been migrated from an earlier version of DB2 and has a blank for FOREIGNKEY in SYSIBM. SYSCOLUMNS. If the DATASUBTYPE command is found for a column, it overrides whatever FOREIGNKEY values exist in the catalog.

- Default: None
- Required: No
- Restrictions: None

Required DATASUBTYPE keywords

TABLECREATOR (*creator-id*)

TABLENAME (*table-name*)

COLUMNNAME (*column-name*)

TABLECREATOR, TABLENAME, and COLUMNNAME identify the source column that will be masked.

- Default: None
- Required: Yes
- Restrictions: None
- Short form: TBCR, TBNM, CLNM

SUBTYPE(B | M | S)

Indicates the data subtype to be assumed as the FOREIGNKEY in SYSCOLUMNS. Valid subtypes are B for BIT, M for MIXED, or S for SBCS.

- Default: S
- Required: No
- Restrictions: None
- Short form: SUBT

HLQDDDF

The HLQDDDF command is optional.

HLQDDDF allows DB2 Cloning Tool Table Space Cloning to pass input and output DDs to ADRDSSU.

HLQDDDF command syntax

HLQDDDF

Required keywords:

DDNAME(*dd1*,...*ddn*)

HLQNAME(*hlq1*)

optional keywords . . .

Optional keywords:

[DIRECTION(IN | OUT)]

HLQDDDF command and keyword definitions

Required and conditionally required keywords are described first, followed by optional keywords

HLQDDDF

The HLQDDDF command is optional. HLQDDDF passes input and output DDs to ADRDSSU.

- Default: None
- Required: Yes
- Restrictions: None

Required HLQDDDF keywords

DDNAME (*vdd1,...ddn*)

DDNAME provides the DD specification to pass to ADRDSSU as an input or output DD.

- Default: None
- Required: Yes
- Restrictions: None

HLQNAME(*hlq*)

The HLQNAME parameter specifies the high level qualifier this command applies to.

- Default: None
- Required: Yes
- Restrictions: None

Optional HLQDDDF keywords

DIRECTION(IN | OUT)

DIRECTION specifies whether this DD name or names is provided to ADRDSSU as an input or output DD.

- Default: OUT
- Required: No
- Restrictions: None
- Short form: DIR

LISTDEF

The LISTDEF command is required. LISTDEF selects all DB2 table spaces and index spaces to be involved in the cloning process.

DB2 Cloning Tool Table Space Cloning's LISTDEF processing differs from IBM's LISTDEF in the following ways:

1. LIST is not supported.
2. Users can not change the name of the LISTDEF DD. It is CKZLSTDF. That DD must point to a single member with the LISTDEF control statements, for one and only one list.
3. The default SQL ID is entered using a DB2 Cloning Tool Table Space Cloning SET command.
4. A detailed explanation of the LISTDEF processing is available when CKZLOG DD is specified for output. This includes a hierarchical list of all table spaces and index spaces and INCLUDE/EXCLUDE processing.

5. LISTDEF output is available in the CKZPRINT DD. It consists of the data sets and the tables referenced.
6. ' /* ' and ' */ ' can be used to enter comments. Note that these comment start and stop commands must be bounded by spaces.
7. Individual INCLUDE/EXCLUDE commands are denoted by a .01, .02, etc. in the LISTDEF Report, such that if LIST074K has one INCLUDE and one EXCLUDE command, they would appear as LIST074K.01 and LIST074K.02. However, note that if the list name is longer than 15 characters, the suffix is not used, as the suffix must fit into the 18 character length of the list name.
8. If an underscore character and wildcards are used to specify the table spaces or index spaces to be selected, DB2 Cloning Tool Table Space Cloning uses an SQL LIKE to find matching DB2 table spaces or index spaces. If an underscore character is used without wildcards to specify the table spaces or index spaces to be selected, DB2 Cloning Tool Table Space Cloning uses an SQL = to locate the table spaces or index spaces.
9. If specifying the HISTORY keyword, it must be specified before the ALL, BASE, LOB, or XML keyword.

Requirements for LISTDEF command syntax

The DB2 Cloning Tool Table Space Cloning LISTDEF syntax is similar to DB2 LISTDEF command syntax. However, DB2 Cloning Tool Table Space Cloning LISTDEF syntax does not make use of continuation characters. Users must enter DB2 Cloning Tool Table Space Cloning LISTDEF keywords in a specified order. The order is as follows:

```
LISTDEF listname
INCLUDE | EXCLUDE
[ TABLESPACES | INDEXSPACES [COPY (YES | NO) ] ]
DATABASE database-name
  [ STOGROUP (stogroupname | stogroupmask) ]
  TABLESPACE database.table-space-name
  INDEXSPACE database.index-space-name
  TABLE creator-id.table-name
  INDEX creator-id.index-name
[ PARTLEVEL( partitionnumber ) ]
[ CLONED(YES | NO) ]
[ RI ]
[ HISTORY ]
[ ALL | BASE | LOB | XML ]
```

For readability, this example shows each keyword on its own line, but this is not required; refer to the examples for variations.

Examples

```
LISTDEF ZTS0405A
  INCLUDE TABLESPACES TABLESPACE ZDSTDB04.ZDSTTS04 RI ALL
  INCLUDE TABLESPACES TABLESPACE ZDSTDB05.ZDSTTS05 RI ALL

LISTDEF ZTN
  INCLUDE TABLESPACES DATABASE ZTNDB
  INCLUDE INDEXSPACES DATABASE ZTNDB
  EXCLUDE TABLESPACES TABLESPACE ZTNDB.ZTN7TSN1
  INCLUDE TABLESPACES TABLESPACE ZTNDB.ZTN5TSN1 PARTLEVEL(0001)
  INCLUDE TABLESPACES TABLESPACE ZTNDB.ZTN5TSN1 PARTLEVEL(0002)
  INCLUDE TABLESPACES TABLESPACE ZTNDB.ZTN5TSN1 PARTLEVEL(0003)
```

LISTDEF command syntax

```
LISTDEF listname
```

Required keywords:

```
LISTDEF listname
INCLUDE | EXCLUDE
DATABASE dbname
| TABLESPACE database.tablespace-name
| INDEXSPACE database.indexspace-name
| TABLE creatorid.table-name
| INDEX creatorid.index-name
optional keywords . . .
```

Optional keywords:

```
[ HISTORY ]
[ ALL | BASE | LOB | XML ]
[ CLONED(YES | NO) ]
[ PARTLEVEL( partitionnumber ) ]
[ RI ]
[ STOGROUP( stogroupname | stogroupmask ) ]
[ TABLESPACES | INDEXSPACES [COPY (YES | NO) ] ]
```

Note: COPY can only be specified with INDEXSPACES; YES | NO is required if COPY is specified.

LISTDEF command and keyword definitions

Required keywords are described first, followed by optional keywords.

LISTDEF *listname*

The LISTDEF command controls the selection of the table spaces and index spaces being copied from the source to the target DB2. It allows selection of individual partitions of a partitioned table space, or LOB columns, and the RI keyword causes it to select all table spaces and index spaces in an RI relationship.

- Default: None
- Required: Yes
- Restrictions: None

Required and Conditionally Required Keywords

INCLUDE

EXCLUDE

The INCLUDE/EXCLUDE parameters determine if the selected table spaces and index spaces are to be added or excluded from the list.

- Default: None
- Required: Yes
- Restrictions: EXCLUDEs should not be followed by an over-riding INCLUDE.

DATABASE *database-name*

TABLESPACE *database.tablespace-name*

TABLE *creator-id.table-name*

INDEXSPACE *database.indexspace-name*

INDEX *creator-id.index-name*

The DATABASE, TABLESPACE, TABLE, INDEXSPACE, and INDEX parameters specify the type of object being named.

- Default: None
- Required: Yes
- Restrictions: None

Optional Keywords

HISTORY

The HISTORY parameter specifies that history tables associated with temporal tables are to be included in the list. HISTORY must be specified before the ALL, BASE, LOB, or XML parameter. When HISTORY is specified, only history tables are processed on that LISTDEF statement. If HISTORY is not specified, no history tables are processed on that LISTDEF statement. To process both history and non-history tables, specify HISTORY and the COPY command parameter ALWAYS-COPY-HISTORY-TABLES(Y).

Object specifications must include the table space, table or index objects, not their corresponding history objects. For example, specify INCLUDE TABLESPACES TABLE *creator.table* HISTORY ALL, not INCLUDE TABLESPACES TABLE *creator.table_hist* HISTORY ALL.

- Default: None
- Required: No
- Restrictions: Valid only for DB2 10 and later subsystems.

ALL BASE LOB XML

ALL specifies that base tables, LOB table spaces and XML table spaces are included.

BASE specifies that only the base table is included.

LOB specifies only LOB table spaces are included. Indexes are included for the selected spaces only.

XML specifies that only XML table spaces are included.

- Default: None
- Required: No
- Restrictions: If BASE, LOB, or XML is specified, the copied target space(s) may be unusable.

CLONED (YES | NO)

The CLONED parameter is used to selectively include or exclude DB2 clone tables.

YES specifies that only table spaces and index spaces that are cloned are to be copied. The base table is always included with the clone table.

NO specifies that only table spaces and index spaces that are NOT cloned are to be copied.

CLONED is optional, but YES or NO is required if CLONED is specified.

- Default: None
- Required: No
- Restrictions: Valid only on DB2 Version 9.1 or later subsystems.

PARTLEVEL(*partitionnumber*)

The PARTLEVEL parameter is used to specify individual partitions to be processed. When using PARTLEVEL for table spaces and index spaces, you must ensure that there is no update activity on partitions that are not included in LISTDEF. Failure to do this may result in a partitioned space where all partitions are not in sync with one another.

- Default: None
- Required: No
- Restrictions: When generating DDL, do not specify PARTLEVEL.

RI The RI parameter specifies that referentially related table spaces and index spaces are to be included in the list.

- Default: None
- Required: No
- Restrictions: None

STOGROUP (*stogroupname* | *stogroupmask*)

The STOGROUP parameter is used to select all objects in all databases in a DB2 storage group. Internally, the LISTDEF statement is replaced by one statement for each database in the DB2 stogroup. STOGROUP % is not valid. Wildcards other than the single % are allowed. For example: INCLUDE INDEXSPACES STOGROUP AB* is allowed.

- Default: None
- Required: No
- Restrictions: Cannot be used when specifying CATALOG-PREFETCH.

TABLESPACES

INDEXSPACES COPY (YES | NO)

The INDEXSPACES and TABLESPACES parameters specify whether this statement pertains to table spaces or index spaces.

COPY can only be specified with INDEXSPACES; YES or NO is required if COPY is specified.

- Default: None
- Required: No
- Restrictions: None

SET

The SET command is required.

SET specifies the local DB2 subsystem and optionally, other job-wide specifications for the source, target, TCP/IP server and report jobs.

SET command syntax

SET

Required keywords:

LOCAL-SSID(*localsubsystem*)
optional keywords . . .

Conditionally Required Keywords:

SET XMLSTRINGDD(*ddname*)

Optional keywords:

[ADVISORY-STATUS-VALUES(*status1,status2,...statusn* | blank)]
 [CONNECT-DB2-ON-CLIENT-CONNECT(Y | N)]
 [DB2-COMMAND-RESPONSE-WAIT]
 [DB2-PLAN(*planname*)]
 [DEFAULT-SQLID(*sqlid*)]
 [IP-VERSION6(Y | N)]


```

[ KEEP-DATABASES-ON-DISCONNECT ( Y | N )
[ MAX-COPY-RC( 0 | 4 | 8 ) ]
[ MAX-RC( 0 | 4 ) ]
[ MAX-SUBTASKS( number of subtasks ) ]
[ MERGE-PRINT( Y | N ) ]
|
[ REBUILD-INDEXES-EXECUTE( Y | N ) ]
[ REMOTE-CONNECT-TYPE( C | D | T ) ]
|
[ REPORT-JOB( Y | N ) ]
[ RESTRICT-STATUS-VALUES( status1,status2,...statusn | blank ) ]
[ SCAN-ONLY( Y | N ) ]
|
[ SOURCE-TCPIP-SERVER-JOB ( Y | N ) ]
[ SOURCE-TCPIP-SERVER-PORT ( port ) ]
|
[ SOURCE-TCPIP-STC-NAME ( started task name ) ]
|
[ SOURCE-CONNECT-DB2-ON-CLIENT-CONNECT ( Y | N ) ]
|
[ SOURCE-IP-VERSION ( Y | N ) ]
|
[ SQL-OUTDD( dd ) ]
|
[ SUBTASK-DATASET-EXTENSIONS( Y | N ) ]
[ SUBTASK-TERMINATION-WAIT ( nnnn ) ]
|
[ TARGET-JOB( Y | N ) ]
|
[ TARGET-JOB-REPAIR-EXECUTE( Y | N ) ]
|
[ TARGET-JOB-REPAIR-SELECT( Y | N ) ]
|
[ TARGET-JOB-REPAIR-TEST( Y | N ) ]
|
[ TCP-SERVER-JOB( Y | N ) ]
[ TCPIP-SERVER-PORT( port ) ]
[ TCPIP-STC-NAME( started task name ) ]
[ TEMPLATE-VARIABLE(variablename1,variablestring1,...
  [ , variablenamen,variablestringn ] ) ]
|
[ USE-RUNTIME-REPOSITORY( Y | N ) ]
|
[ UTILITY-COMMAND-EXECUTE-PERCENT(percentage per call) ]

```

SET command and keyword definitions

Required and conditionally required keywords are described first, followed by optional keywords.

SET The SET command is required. It specifies the source DB2 subsystem and, optionally, other job-wide specifications for each of the three jobs (source, target, and TCP/IP server).

- Default: None
- Required: Yes
- Restrictions: None

Required and conditionally required SET keywords

Note: The Jobs entry in this parameter list displays the jobs in which you are allowed to enter the parameter with a SET command.

LOCAL-SSID(*localsubsystem*)

The LOCAL-SSID parameter provides the name of the local DB2 subsystem. This is the source subsystem for the source job and the target subsystem for the target and TCP/IP server jobs.

- Default: None
- Required: Yes
- Restrictions: None
- Short form: LSSID
- Jobs: Source, TCP/IP server

XMLSTRINGDD(*ddname*)

The XMLSTRINGDD parameter is created by the source job. It provides the ddname to the target job to indicate that there is an XMLSTRING DD passed to the target job.

- Default: None
- Required: Required if one or more XML columns are present in the source subsystem tables and will be copied to the target subsystem.
- Restrictions: None
- Short form: XMLSDD
- Jobs: None (created by DB2 Cloning Tool Table Space Cloning source job)

Optional SET keywords

ADVISORY-STATUS-VALUES (*status1,status2,...statusn* | **blank**)

The ADVISORY-STATUS-VALUES parameter allows you to specify which status values are to be checked for table spaces and index spaces before the copies are performed. When a specified status is detected, the space is marked as mismatched and a warning message is issued. If ALLOW-COPY-ON-MISMATCH and MAX-RC(4) are in effect, the copy may proceed. Otherwise, the copy is not allowed for all affected data sets.

This keyword determines the content of the DB2 DISPLAY command:

```
-DISPLAY DATABASE(dbname) SPACENAM(*) ADVISORY(s1,s2,...sn)
```

where *dbname* is one of the databases that is being copied and *s1,s2,...,sn* are valid status values.

The following are valid values for *status1...statusn*:

- ICOPY: informational COPY-pending
- AUXW: auxiliary warning
- ARBDP: advisory REBUILD-pending
- AREO*: advisory REORG-pending
- AREOR: advisory REORG-pending (DB2 10 and later)
- Blank: If SET ADVISORY-STATUS-VALUES() is used, the ADVISORY portion of the DISPLAY command is turned off.

One DISPLAY command is generated for each database that has one or more spaces that are to be copied. Both the source and target subsystems are checked for the specified status values. If a default value or a specified status value does not apply to the DB2 version, it is deleted from the command.

- Default: ARBDP, AREO*, AREOR (for DB2 10 and later), and AUXW are used if no SET ADVISORY-STATUS-VALUES command is found.
- Required: No
- Restrictions: None
- Short form: ADVIS
- Jobs: Source

CAUTION:

If objects are that have been migrated from DB2 V7 have had columns added using ALTER TABLE ADD COLUMN, the AREO* status may not be displayed. In this case, you must REORG the table spaces before using DB2 Cloning Tool Table Space Cloning to copy them. Failure to perform this REORG may result in inaccessible target data.

CONNECT-DB2-ON-CLIENT-CONNECT(Y | N)

The CONNECT-DB2-ON-CLIENT-CONNECT parameter can be used to allow the TCP/IP server job to connect to DB2 only when a client connect

| occurs. When this parameter is set to Y, the TCP/IP server disconnects
| from DB2 when the client disconnects. This allows the TCP/IP server to be
| run as a started task; the TCP/IP server job can be started at IPL and left
| running. If a DB2 connect error occurs when the source job is running the
| TCP/IP server job, the TCP/IP server job ends with a return code of 8, and
| must be restarted.

- Default: N
- Required: No
- Restrictions: None
- Short form: CDOCC
- Jobs: TCP/IP server

DB2-COMMAND-RESPONSE-WAIT

The DB2-COMMAND-RESPONSE-WAIT is the number of seconds DB2 Cloning Tool Table Space Cloning waits for a space to go from STOPP to STOP status after the STOP DB2 command is issued. The range is 1 - 999999 seconds.

- Default: 60 seconds.
- Required: No
- Restrictions: None
- Short form: DB2CWAIT
- Jobs: Source, TCP/IP server

DB2-PLAN(*plan_name*)

The DB2-PLAN parameter specifies the DB2 plan name for source and target DB2 subsystem connects.

Note: The target job inherits the SET value from the source job.

- Default: The value that is entered in the PARMLIB.
- Required: No
- Restrictions: None
- Short form: PLAN
- Jobs: Source, TCP/IP server

DEFAULT-SQLID(*sqlid*)

The DEFAULT-SQLID parameter, when specified, causes DB2 Cloning Tool Table Space Cloning to apply this creator ID to all tables and indexes in LISTDEF without an explicit creator ID.

- Default: None
- Required: No
- Restrictions: None
- Short form: DEFSQLID
- Jobs: Source

IP-VERSION6(Y | N)

The IP-VERSION6 parameter specifies the version of TCP/IP used on the TCP/IP server.

N (the default) specifies IPv4.

Y specifies IPv6.

- Default: N
- Required: No

- Restrictions: None
- Short form: IPV6
- Jobs: Source, TCP/IP server

KEEP-DATABASES-ON-DISCONNECT (Y | N)

When using a TCP/IP server, KEEP-DATABASES-ON-DISCONNECT (Y) can be used to populate the server with prefetched target objects. Populate the server cache of database objects by running source jobs using PGM(NONE). The TCP/IP server job saves objects as long as it has enough storage to hold them. If a target object changes before the source COPY job is run, the TCP/IP server must be canceled and rerun as the cached definition could be wrong. When using the TARGET-PREFETCH -DATABASE-LIST, every object in the list of databases is cached.

- Default: N
- Required: No
- Restrictions: None
- Short form: KDBOD
- Jobs: TCP/IP server

MAX-COPY-RC (0 | 4 | 8)

Specifies the maximum return code for data set copy. When the specified return code is exceeded, the job ends in error. Use this parameter in the source job to override the default or PARMLIB specification. This parameter allows copies to continue even with warnings (value 4) or errors (value 8). Thus, all copies that can complete while all spaces are stopped completes.

- Default: PARMLIB specification for MAX_COPY_RC.
- Required: No
- Restrictions: None
- Short form: MXCPRC
- Jobs: Source

MAX-RC (0 | 4)

Specifies the maximum job return code. When the specified return code is exceeded, the job ends in error. Use this parameter in the source or TCP/IP jobs to override the default or PARMLIB specification. This parameter determines whether the job continues with warnings or terminates.

- Default: PARMLIB specification for MAX_RC.
- Required: No
- Restrictions: None
- Short form: MXRC
- Jobs: Source, TCP/IP server

MAX-SUBTASKS (*number of subtasks*)

Specifies the number of subtasks to start. Use this parameter in the source job to override the default or PARMLIB specification. This parameter can be used to reduce the elapsed time of long-running source and target jobs by allowing multiple I/O operations to run concurrently. DB2 processing is impacted in the source job and VSAM processing in the target job. As the target job must access every page, it is particularly useful. Refer to the MAX_SUBTASKS description in the topic "CKZINI keyword syntax and descriptions" on page 577 for more recommendations for the value of this parameter.

Note: The target job inherits the SET value from the source job. To use different values for each job, use two PARMLIB members: one for the source job and one for the target job, each with its own MAX_SUBTASKS value.

When using a high value for subtasks, specify SUBTASK-TERMINATION-WAIT(*nnn*) in the source job. *nnn* should be enough seconds to allow all subtasks to come down. If it is not large enough, an A03 abend occurs. The SUBTASK-TERMINATION-WAIT command will also be used in the target job.

- Default: PARMLIB specification for MAX_SUBTASKS.
- Required: No
- Restrictions: None
- Short form: MXSUBT
- Jobs: Source

MERGE-PRINT (Y | N)

This parameter allows message output from CKZPRINT and CKZLOG to be combined into CKZPRINT. It can be specified in the source or TCP/IP server jobs and is useful when you are investigating a problem or sending documentation to IBM Software Support.

Note: The target job inherits the SET value from the source job.

- Default: N
- Required: No
- Restrictions: None
- Short form: MERGEP
- Jobs: Source, TCP/IP server

REBUILD-INDEXES-EXECUTE (Y | N)

When set to Y, the REBUILD-INDEXES-EXECUTE parameter indicates that index rebuilds are to be submitted as part of the target job. If you specify REBUILD-INDEXES-EXECUTE(Y), you must also provide a DD name in the COPY command TARGET-JOB-INDEX-REBUILD-DDN.

- Default: N
- Required: No
- Restrictions: None
- Short form: RBIXX
- Jobs: Source

REMOTE-CONNECT-TYPE (C | D | T)

The REMOTE-CONNECT-TYPE parameter is used to specify a particular connection type for the remote (target) subsystem. Specify C to indicate CAF, D for DDF, and T for TCP/IP.

When the source and target DB2 subsystems have the same name, specify T to indicate that the same named subsystem is used for both source and target. This allows the target subsystem to be connected via TCP/IP.

If the keyword is not included, connections are attempted in this order: CAF, DDF, and TCP/IP. This command is the preferred method of specifying all DDF and TCP/IP connections. For example, if no keyword is specified when TCP/IP is used and DDF fails, error messages are printed for the DDF failure. If SET REMOTE-CONNECT-TYPE(T), then DDF is not attempted and thus no error message will be issued.

- Default: If this keyword is not specified, connections are attempted in this order: C, D, T.
- Required: No
- Restrictions: None
- Short form: RCTYP
- Jobs: Source

REPORT-JOB (Y | N)

This parameter is used to determine whether a report based on data in the target job runtime repository is to be output. The report job runs as a separate job. A sample job with instructions is contained in SCKZJCL (CKZREPJB).

- Default: N
- Required: No
- Restrictions: None.
- Short form: REPJOB
- Jobs: Report

RESTRICT-STATUS-VALUES (*status1,status2,...statusn* | **blank**)

The RESTRICT-STATUS-VALUES parameter allows you to specify which status values are to be checked for table spaces and index spaces before the copies are performed. When a specified status is detected, the space is marked as mismatched and a warning message is issued. If ALLOW-COPY-ON-MISMATCH and MAX-RC(4) are in effect, the copy proceeds. Otherwise, the copy is not allowed for all affected data sets.

This keyword determines the content of the DB2 DISPLAY command:
 -DISPLAY DATABASE(*dbname*) SPACENAM(*) RESTRICT(*s1,s2,...sn*)

where *dbname* is one of the databases that is being copied and *s1,s2,...,sn* are valid status values.

The following are valid values for *status1...statusn*:

- ACHKP: auxiliary CHECK-pending
- CHKP: CHECK-pending
- COPY: COPY-pending
- GRECP: group buffer pool RECOVER-pending
- LPL: logical page list entries
- RBDP: REBUILD-pending
- RECP: RECOVER-pending
- REORP: REORG-pending
- RO: Read-only mode
- STOP: Stopped objects, including the restricted states STOP, STOPE, STOPP, and LSTOP
- UT: Utility access mode
- UTRO: Serialized for utility access and available for read-only access
- UTRW: Serialized for utility access and available for read-write access
- UTUT: Serialized for utility access and unavailable
- UT*: Any utility access mode: UT, UTRW, UTRO, or UTUT
- WEPR: Displays write error page range information
- Blank: If SET RESTRICT-STATUS-VALUES() is used, the RESTRICT portion of the DISPLAY command is turned off.

One DISPLAY command is generated for each database that has one or more spaces that are to be copied. Both the source and target subsystems are checked for the specified status values. If a default value or a specified status value does not apply to the DB2 version, it is deleted from the command.

- Default: ACHKP, CHKP, GRECP, RBDP, RECP, REORP, and UT* are used if no SET RESTRICT-STATUS-VALUES command is found.
- Required: No
- Restrictions: None
- Short form: RESTR
- Jobs: Source

SCAN-ONLY(Y | N)

The SCAN-ONLY parameter controls the target job updating OBIDs and LOGRBAs in the target job.

If Y, then all the target data sets are read, and page change messages are output in the log; however the page sets are not written.

- Default: N
- Required: No
- Restrictions: This parm is set to N by DB2 Cloning Tool Table Space Cloning when the CKZSYNC DD is updated in the source job. This parm may be changed to Y before running the target job.
- Short form: SCANO
- Jobs: None

SOURCE-TCPIP-SERVER-JOB (Y | N)

This parameter distinguishes the source TCP/IP server job from the source job and target job.

- Default: N
- Required: No
- Restrictions: Specify in the SOURCE-SERVER job
- Short form: STCPJOB
- Jobs: Source, TCP/IP server

SOURCE-TCPIP-SERVER-PORT (port)

This parameter allows you to specify the connection port that the source TCP/IP server is to use. Use this parameter to override the default and/or PARMLIB specification. Valid values are 1 to 65535.

- Default: 5099
- Required: No
- Restrictions: Specify in the SOURCE or SOURCE-SERVER jobs
- Short form: STSPORT
- Jobs: Source, TCP/IP server

SOURCE-TCPIP-SERVER-IP (ip)

This parameter specifies the connection IP address for the LPAR on which the source TCP/IP server is executing.

- Default: None
- Required: No
- Restrictions: Specify in the SOURCE job
- Short form: SSIP

- Jobs: Source, TCP/IP server

SOURCE-TCPIP-STC-NAME (*started task name*)

This parameter allows you to specify the name of the source TCP/IP address space that the TCP/IP source server connects to. Specify a valid z/OS started task name.

- Default: TCPIP
- Required: No
- Restrictions: Specify in the SOURCE-SERVER jobs
- Short form: STSNAME
- Jobs: Source, TCP/IP server

SOURCE-CONNECT-DB2-ON-CLIENT-CONNECT (Y | N)

The SOURCE-CONNECT-DB2-ON-CLIENT-CONNECT parameter can be used to allow the source TCP/IP server job to connect to DB2 only when a client connect occurs. When this parameter is set to Y, the source TCP/IP server disconnects from DB2 when the client disconnects. This allows the source TCP/IP server to be run as a started task; the source TCP/IP server job can be started at IPL and left running. If a DB2 connect error occurs when the source job is running the source TCP/IP server job, the source TCP/IP server job ends with a return code of 8, and must be restarted.

- Default: N
- Required: No
- Restrictions: Specify in the SOURCE-SERVER job
- Short form: SCDOCC
- Jobs: TCP/IP server

SOURCE-IP-VERSION (Y | N)

The SOURCE-IP-VERSION parameter specifies the version of TCP/IP used on the source TCP/IP server. N (the default) specifies IPv4. Y specifies IPv6.

- Default: N
- Required: No
- Restrictions: Specify in the SOURCE-SERVER job
- Short form: SIPV6
- Jobs: TCP/IP server

SQL-OUTDD(*dd*)

The SQL-OUTDD parameter provides the name of the SQL data set to the target job. It is required if ALTER TABLE SQL is needed to synchronize identity column start values. This parameter is set by DB2 Cloning Tool Table Space Cloning when the CKZSYNC DD is updated in the source job.

- Default: None
- Required: No. Needed only if ALTER TABLE SQL is needed to synchronize identity column start values.
- Restrictions: It is not coded by the user, but inserted by DB2 Cloning Tool Table Space Cloning into the CKZSYNC DD member by the source job.
- Short form: SQLDD
- Jobs: None

SUBTASK-DATASET-EXTENSIONS(Y | N)

The SUBTASK-DATASET-EXTENSIONS parameter allows data set extents

| to be processed in any available subtask. When a significant number of
| extents may be processed by the target job, setting this parameter to Y
| might improve the total elapsed time of the target job. If DB2 START
| commands are enabled for target objects, the objects will be started when
| the last extent of an object has been processed, regardless of the subtask
| used. The number of subtasks (set via PARMLIB parameter
| MAX_SUBTASKS or COPY command MAX-SUBTASKS parameter) must
| be greater than one for this to be a useful feature.

- | • Default: N
- | • Required: No
- | • Restrictions: Not allowed with PGM(SRCIMCPY) or
| USE-RUNTIME-REPOSITORY(Y), when using data masking or log apply
| functionality, or when DB2 STOP commands may appear in the target
| job (refer to the COPY parameter AUTO-STOP-TARGET-SPACE or
| PARMLIB parameter AUTO_STOP_TARGET_SPACE).
- | • Short form: SUBDX
- | • Jobs: Source

SUBTASK-TERMINATION-WAIT(*nnnn*)

The SUBTASK-TERMINATION-WAIT keyword allows you to specify a
wait time for all subtasks to end before exiting. In the target job, a large
data set being updated with new OBIDs can take a long time to process. In
this case, DB2 Cloning Tool Table Space Cloning may exit before the
subtask ends, resulting in an A03 ABEND. Valid values are 0 to 9999
seconds; 0 indicates that DB2 Cloning Tool Table Space Cloning is to wait
for all subtasks to end before exiting.

- | • Default: 60
- | • Required: No
- | • Restrictions: None
- | • Short form: STWAIT
- | • Jobs: Source, target, TCP/IP server

TARGET-JOB(*Y | N*)

The TARGET-JOB parameter distinguishes the target job from the source
job and the TCP/IP server job. This parm is set to Y by DB2 Cloning Tool
Table Space Cloning when the CKZSYNC DD is updated in the source job.

The parm value is N (the default) if the source job or TCP/IP server job,
and Y if the target job.

- | • Default: N
- | • Required: No
- | • Restrictions: None
- | • Short form: TRGJOB
- | • Jobs: None.

TARGET-JOB-REPAIR-EXECUTE(*Y | N*)

| This parameter is used when generating REPAIR utility jobs with job
| templates. Specify TARGET-JOB-REPAIR(Y) to generate REPAIR jobs to
| detect and correct catalog or DB2 version inconsistencies. This can occur
| when DB2 objects are copied from one subsystem to another. When
| TARGET-JOB-REPAIR-EXECUTE is set to Y, REPAIR jobs are submitted by
| DB2 Cloning Tool Table Space Cloning near the end of the target job. If the
| parameter is set to N or not included, REPAIR jobs are not submitted.

The inconsistencies processed are record format (BRF vs. RRF), actual page format vs. catalog column RBA_FORMAT, HASHDATAPAGES, and versioned objects. Record format, page format, and HASHDATAPAGES processing only apply to DB2 V11 and may only be run on table spaces. Versioned object processing applies to all DB2 versions; DB2 V9 and V10 REPAIR VERSIONS can be run on table spaces and index spaces, while DB2 V11 REPAIR VERSIONS can be run on table spaces only.

- Default: N
- Required: No
- Restrictions: None
- Short form: TRGRX
- Jobs: Source.

TARGET-JOB-REPAIR-SELECT (Y | N)

This parameter is used when generating REPAIR utility jobs with job templates. When TARGET-JOB-REPAIR-SELECT is set to Y, the REPAIR utility is processed at the end of the target job only for those table spaces matching the criteria defined for the DB2 version. If the parameter is set to N, no table spaces are processed. If the parameter is not present, no table spaces are processed.

- Default: N
- Required: No
- Restrictions: None
- Short form: TRGRS
- Jobs: Source.

TARGET-JOB-REPAIR-TEST(Y | N)

This parameter is used when generating REPAIR utility jobs with job templates. When TARGET-JOB-REPAIR-TEST is set to Y, the REPAIR job is submitted by DB2 Cloning Tool Table Space Cloning, but mismatch information is not corrected in the catalog. The mismatch information that results in non-zero return codes is reported in the job output.

- Default: N
- Required: No
- Restrictions: None
- Short form: TRGTS
- Jobs: Source.

TCPIP-SERVER-PORT (*port*)

This parameter allows you to specify the port that the TCP/IP server is to use to wait for requests from the source job. It may also be used in the source job. Use this parameter to override the default and/or PARMLIB specification. Valid values are 1 to 65536.

Note: The source job port specified with this SET command shall be overridden if specified using SERVER-PORT on the COPY command. This port and the one specified for the TCP/IP server must match.

- Default: PARMLIB specification for TCPIP_SERVER_PORT
- Required: Yes, in the TCP/IP server job.
- Restrictions: None
- Short form: TSPORT
- Jobs: Source, TCP/IP server

TCP-IP-STC-NAME (*started task name*)

This parameter allows you to specify the name of the TCP/IP address space that the TCP/IP server and source job connect to. This parameter can be specified in source and TCP/IP server jobs and can be used to override the default and/or PARMLIB specification. Specify a valid z/OS started task name.

- Default: PARMLIB specification for TCPIP_STC_NAME
- Required: No
- Restrictions: None
- Short form: TSNAME
- Jobs: Source, TCP/IP server

TCP-SERVER-JOB (**Y | N**)

The TCP-SERVER-JOB parameter distinguishes the TCP/IP server job from the source job and the target job.

N (the default) if the source job or target job.

Y if the TCP/IP server job.

- Default: N
- Required: No
- Restrictions: None
- Short form: TCPJOB
- Jobs: TCP/IP server

TEMPLATE-VARIABLE (*variablename1,variablestring1, ... [, variablenamen,variablestringn]*)

The TEMPLATE-VARIABLE parameter allows a variable to be set for the life of the source job. The maximum variable name length is 8 characters and the maximum variable length is 68. When making a substitution, the variable expansion cannot extend the line past column 71. These are straight substitution variables, i.e., no processing is controlled by these values.

The variable name must begin with && and the variable string cannot have embedded blanks, an ampersand or field terminator (.). With SET VARIABLES, multiple copy commands to multiple targets can be built with a single source job. For example, each template could have a separate temporary variable for the high level qualifier following the RENAMEU command. If copied target data sets must be accessible, subsequent source jobs can be run against secondary targets without copying the data [PGM(NONE)], to output a SYNCDB2 file to use for target jobs to correct OBIDs.

- Default: N
- Required: No
- Restrictions: None
- Short form:
- Jobs: Source

USE-RUNTIME-REPOSITORY(**Y | N**)

This parameter is used to determine if all the data sets in the target job are processed. It utilizes a target job runtime repository that keeps track of target jobs and all the data sets that are processed by the target jobs. This repository can allow the failed target job to be restarted and skip any target data sets that have been processed. If USE-RUNTIME-REPOSITORY

| is set to Y and the target job produces a non-zero return code, when the
| target job is re-run it will process only the data sets that were not
| previously started. If USE-RUNTIME-REPOSITORY is set to N, the target
| job processes all data sets in the target job.

| For USE-RUNTIME-REPOSITORY(Y), you must:

- | 1. Add these DDs to source and target jobs:
 - | • //CKZRRJOB DD DISP=OLD,DSN=&h1q.RRJOB
 - | • //CKZRRDSN DD DISP=OLD,DSN=&h1q.RRDSN
- | 2. Allocate the DDs; allocation instructions are provided in
| SCKZJCL(CKZRREP).
- | 3. Add USE-RUNTIME-REPOSITORY(Y) to the SET command in the
| source job.
 - | • Default: N
 - | • Required: No
 - | • Restrictions: USE-RUNTIME-REPOSITORY(Y) is not valid with
| PGM(SRCIMCPY).
 - | • Short form: USERR
 - | • Jobs: Source

| **UTILITY-COMMAND-EXECUTE-PERCENT**(*percentage per call*)

| This parameter specifies the percentage of objects or data sets that are
| eligible for a particular DB2 utility to be run in a single call to that utility.
| Specify a percentage of the total data sets or objects from 1 - 50 inclusive.
| For example, if UTILITY-COMMAND-EXECUTE-PERCENT is set to 20,
| and there are 500 table spaces eligible for a QUIESCE, then five QUIESCE
| calls would be submitted with 100 table spaces in each call. The default
| specifies one utility call for all eligible spaces.

| **Note:** Currently, the only utility this parameter can be specified for is the
| QUIESCE command when submitted during log apply processing.

- | • Default: 0 (all eligible spaces are submitted with a single call)
- | • Required: No
- | • Restrictions: None
- | • Short form: UTCXP
- | • Jobs: Source

Chapter 25. Tools Customizer reference

Before you use Tools Customizer, you should understand the Tools Customizer terminology and the data sets that Tools Customizer uses during customization.

Tools Customizer terminology and data sets

Before you use Tools Customizer, you should understand the Tools Customizer terminology and the data sets that Tools Customizer uses during customization.

Tools Customizer terminology

Tools Customizer uses several unique terms that you should be familiar with before you begin to use Tools Customizer.

Products and components

How an IBM Tool is packaged determines whether it is referred to as a product or as a component in the Tools Customizer documentation and interface. An IBM Tool that is ordered as a stand-alone entity (that is, not as part of a solution pack) is referred to as a product. An IBM Tool that is part of a solution pack is referred to as a component. Some IBM Tools are available in both formats; therefore, the same IBM Tool can be referred to as a product or as a component depending on how it is packaged.

DB2 entry

You can customize DB2 Cloning Tool on one or more DB2 entries. A DB2 entry can be any of the following items:

DB2 subsystem

A distinct instance of a relational database management system (RDBMS) that is not part of a data sharing group. An example of a DB2 subsystem name is DB01.

DB2 group attach name

The name that is used by the TSO/batch attachment, the call attachment facility (CAF), DL/I batch, utilities, and the Resource Recovery Services attachment facility (RRSAF) as a generic attachment name. An example of a group attach name is DSG1.

DB2 data sharing member

A DB2 subsystem that is assigned by the cross-system coupling facility (XCF) to a data sharing group. An example of a DB2 data sharing member name is DB02.

Tools Customizer maintains the following lists of DB2 entries:

Associated list

The list of DB2 entries that are associated with DB2 Cloning Tool. If the product to be customized requires DB2 entries, you can customize DB2 Cloning Tool only on DB2 entries that are in the associated list. When you customize DB2 Cloning Tool, this list is displayed in the DB2 Entries, Associations, and Parameter Status section of the Customizer Workplace panel.

You can add and copy DB2 entries to the associated list. When you add or copy DB2 entries to the associated list, the entries are associated with DB2 Cloning Tool.

Master list

The list of all DB2 entries that are defined but are not associated with DB2 Cloning Tool. Tools Customizer obtains information about these DB2 entries either from entries that were created manually or from the customizations of other products that were discovered. If you remove a DB2 entry from the associated list, the DB2 entry is added to the master list. When you create a new DB2 entry, it is added to the master list, and when you associate the new entry with DB2 Cloning Tool, it is removed from the master list and added to the associated list. The master list is displayed on the Associate a DB2 Entry for Product panel.

If the associated list does not have the DB2 entries on which you want to customize DB2 Cloning Tool, you can associate existing entries from the master list to the associated list.

You can create new DB2 entries and copy existing entries to the master list.

High-level qualifier

The high-level qualifier is considered to be all of the qualifiers except the lowest level qualifier. A high-level qualifier includes a mid-level qualifier.

Product parameters

Parameters that are specific to DB2 Cloning Tool. These parameters are defined by DB2 Cloning Tool and are stored in a data member that is defined by DB2 Cloning Tool.

DB2 parameters

Parameters for a DB2 entry. These parameters are defined by Tools Customizer and are stored in a DB2 parameter data member.

Status type

Product, LPAR, and DB2 entry status type

After you specify the product that you want to customize, the product, the LPAR, and the DB2 entries have a status. The status is partly based on whether required parameters are defined. For some products, LPAR parameters or DB2 parameters might not be required. In these cases, the status is Not Required.

To customize DB2 Cloning Tool, all of the required parameters must be defined.

If required parameters for the the product parameters or DB2 parameters are not defined, the status of the parameters is Incomplete. Define values for parameters by manually editing them or by generating the customization jobs and specifying values for all of the required parameters that are displayed on the panels.

When values for all of the required parameters are defined, the status is Ready to Customize. Customization jobs can be generated only when all of the required parameters are defined and the status is Ready to Customize or Customized for the product parameters and DB2 parameters for the DB2 entries on which DB2 Cloning Tool will be customized.

The following table shows the meaning of the status types. Each status is defined differently for each type of parameter.

Table 66. Status types for the product, the LPAR, and the DB2 entries

Status	Product	LPAR	DB2 entries
Incomplete	The required product parameters are not defined, or the required product parameters are defined but LPAR parameters, DB2 parameters, or both are not defined.	The required parameters are not defined.	The required parameters are not defined.
Discovered	The product parameter definitions were discovered by using the product Discover EXEC.	N/A	N/A
Ready to Customize	The required product, LPAR, and DB2 parameters are defined, the status is Ready to Customize or Customized for the LPAR and at least one associated DB2 entry. You can generate the customization jobs.	The required LPAR parameters are defined or LPAR parameters are not required.	The required DB2 parameters are defined or DB2 parameters are not required.
Customized	The jobs are customized on the local LPAR.	The jobs are customized for the product or for all of the associated DB2 entries on the local LPAR.	The jobs are customized for the DB2 entry.
Errors in Customization	N/A	N/A	Errors occurred while the customization jobs were being generated.
Not Required	N/A	LPAR parameters are not required.	DB2 parameters are not required.

Related tasks:

“Creating and associating DB2 entries” on page 58

You can create new DB2 entries and associate them with DB2 Cloning Tool.

“Copying DB2 entries” on page 66

You can copy associated and not associated DB2 entries to other DB2 entries or to new DB2 entries.

“Removing DB2 entries” on page 67

You can remove DB2 entries from the associated list.

Data sets that Tools Customizer uses during customization

Tools Customizer uses several unique data sets during the customization process. Familiarize yourself with these data sets before you begin to use Tools Customizer.

Several different data sets are required to customize DB2 Cloning Tool with Tools Customizer. These data sets are supplied by DB2 Cloning Tool, supplied by Tools Customizer, or allocated by Tools Customizer.

DB2 Cloning Tool provides the following data sets:

Metadata library

Contains the metadata for the product to be customized. Tools Customizer uses the metadata to determine which tasks, steps, and parameters to display on the Product Parameters panel, the LPAR Parameters panel, and the DB2 Parameters panel. This data set also contains the templates that Tools Customizer uses to generate the customization jobs.

The metadata library naming convention is *high_level_qualifier.SCKZDENU*, where *high_level_qualifier* is all of the segments of the data set name except the lowest-level qualifier.

You specify the metadata library on the Specify the Metadata Library panel. READ access to this data set is required.

Discover EXEC library

Contains the DB2 Cloning Tool Discover EXEC. When you customize DB2 Cloning Tool, you can use the Discover EXEC to automatically retrieve and store product information, such as parameter values from an already customized product. Tools Customizer saves the discovered information in the data store.

The default name of the data set is the high-level qualifier for the metadata library plus a lowest-level qualifier. For DB2 Cloning Tool, the lowest-level qualifier is SCKZDENU. You can change the default value on the Discover Customized Product Information panel. EXECUTE access to this data set is required.

Tools Customizer provides the following data sets:

Tools Customizer metadata library

Contains the metadata for the DB2 and LPAR parameters that are required to customize DB2 Cloning Tool. Tools Customizer uses the metadata to determine which parameters to display on the DB2 Parameters panel and the LPAR Parameters panel. In addition, Tools Customizer uses information in the metadata library to determine whether additional DB2 and LPAR parameters need to be displayed on these panels. As you customize different products, different DB2 and LPAR parameters might need to be defined.

The default name of the data set is DB2TOOL.CCQ110.SCCQDENU. You can change the default value on the Tools Customizer Settings panel. READ access to this data set is required.

Tools Customizer table library

Stores information about jobs that are customized. Job information that is stored includes a description of the job, its member name and template name, the SSID, group attach name, and when the job was generated.

The default name of the data set is DB2TOOL.CCQ110.SCCQTENU. WRITE access to this data set is required.

Tools Customizer requires that the following data sets exist during the customization process. If the data sets do not exist, Tools Customizer automatically allocates them.

Discover output data set

Contains the output that is generated when you run the DB2 Cloning Tool Discover EXEC. The DB2 Cloning Tool Discover EXEC retrieves the metadata and values for the parameters from a previous customization of DB2 Cloning Tool.

The default name of the data set is DB2TOOL.CCQ110.DISCOVER. You can change the default value on the Tools Customizer Settings panel or the Discover Customized Product Information panel. WRITE access to this data set is required.

Data store data set

Contains product, LPAR, and DB2 parameter values, and DB2 entry associations. Tools Customizer uses this data set to permanently store all information that is acquired about the product, DB2 subsystems or data sharing groups, and LPAR when you customize products on the local LPAR.

The default name of the data set is DB2TOOL.CCQ110.DATASTOR. You can change the default value on the Tools Customizer Settings panel. WRITE access to this data set is required.

Customization library

Contains the customization jobs that Tools Customizer generates for DB2 Cloning Tool.

Tools Customizer checks whether a customization library name was specified for more than one instance of the same version of the same product. If the same customization library name is specified for more than one product of the same version, the CCQD123E message is issued to prevent you from overwriting previously generated customization jobs. Ensure that you specify unique qualifier for the customization library for each instance of the product.

To customize DB2 Cloning Tool, submit the members of the data set in the order in which they are displayed on the Finish Product Customization panel.

The data set naming convention is *hlq*.\$LPAR_name\$.xyzvrm, where:

- *hlq* is the value of the **Customization library qualifier** field on the Tools Customizer Settings panel (CCQPSET)
- *LPAR_name* is the four-character LPAR name
- *xyzvrm* is the three-letter product identifier with the version, release, and modification level

For example, the data set name might be DB2TOOL.PRODUCT.CUST.\$MVS1\$.XYZ410.

WRITE access to this data set is required.

Tools Customizer allocates the data sets for the discover output, the data store, and the customization library with the attributes that are shown in the following table:

Table 67. Data set attributes for allocating the Discover output, data store, and customization library data sets

Data set	Organization	Record format	Record length	Block size	Data set name type
Discover output data set	PO	Variable block	16383	32760	LIBRARY
Data store data set	PO	Variable block	16383	32760	LIBRARY
Product customization library	PO	Fixed block	80	32720	LIBRARY

Restrictions:

- Multiple users cannot simultaneously share the discover output data set, data store data set, Tools Customizer metadata library, and metadata library.
- You cannot share the data store data set across multiple LPARs with shared DASD or copy the data store data set to another LPAR. Tools Customizer creates many cross-references between product and DB2 associations. Therefore, if you share or copy the data store data set, member names that are empty or that do not exist might be generated.

Chapter 26. Reference

Reference information supports the tasks that you must complete to install, customize, and use DB2 Cloning Tool.

Cloning scenarios

This topic provides examples of potential cloning scenarios.

These scenarios are intended to be used to assist you in planning your cloning methodology. There are many scenarios that can be achieved. If you do not see a scenario in this topic that meets your requirements, contact IBM Software Support.

Volume cloning using an interim set of volumes

Volumes may be cloned using an interim set of volumes.

DB2 Cloning Tool supports this process, but requires knowledge of the original source and final target volumes. The procedure for this type of cloning is:

- The source volumes (referred to as set A) are copied to the interim volumes (set B).
- The interim volumes (set B) are copied to the target volumes (set C).
- The data sets on the target volumes (set C) are renamed.

For this procedure, two COPY commands are used. Each COPY command must be in its own JCL step.

- The first COPY command copies the source volumes to the interim volumes and backs up the source ICF catalogs.
- The second COPY command copies the interim volumes to the target volumes and does not back up the ICF catalogs.

An additional DD must be added to the COPY steps. The DD name is VOLPLIST and it must have attributes of RECFM=FB,LRECL=80. The data set allocated by the VOLPLIST DD will be used by a subsequent step/job.

Step overview

This section summarizes the steps you need to follow to perform volume cloning.

1. Run COPY of volume set A to volume set B using a USERCATALOGS keyword and including the VOLPLIST DD. The ICF catalog is backed up and the journal file that will be used for the RENAME and DB2 steps will be created. The journal contains volume pairs set A and set B. The VOLPLIST data set created in this run contains the volume pairs from this copy (set A and set B) that will be read in step 3.
2. Run COPY of volume set B to volume set C using NOUSERCATALOGS keyword and including the VOLPLIST DD. The ICF catalog is not backed up and the journal file created in this job will not be further used. The VOLPLIST data set created in this run contains the volume pairs from this copy (set B and set C) and will be read in step 3.
3. Run CKZRNTGT with the VOLPLIST from step 1 on the CKZIN DD and the VOLPLIST from step 2 on the NUCIN DD. DB2 Cloning Tool requires the

volume pairs from set A (source) and set C (final target). This step reads the VOLPLIST data set created from the first COPY in step 1 that references set A and set B, and reads the VOLPLIST data set created from the second copy that references set B and set C. It matches up the volume pairs and creates a new temporary data set with the correct volume pairs from set A and set C.

4. Run VOLOPTIONS using NEWTGT data set as input to NEWTARGETS-DDN (temporary data set created in previous step). This job reads in the temporary data set created in step 3 and is used to update the journal data set created in step 1 to point to volume pairs set A and set C.
5. Run RENAME for volume set C.

Note: Steps 1, 4, and 5 use the same journal data set. Step 2 uses a different journal data set.

Example

In the example steps that follow, the following items are used:

- The source volumes are SRC001 and SRC002.
- The interim volumes are INT001 and INT002.
- The target volumes are TGT001 and TGT002

Step 1: Copy SRCxxx to INTxxx and back up the source ICF catalogs

Partial JCL for this step follows:

```
//S1      EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI  DD DISP=SHR,DSN=HLQ?.SCKZPARM(CKZINI)
//CKZPRINT DD SYSOUT=*
//JOURNAL DD DSN=HLQ?.JRNL,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          RECOG=KS,KEYLEN=64,KEYOFF=0,
//          LRECL=600,SPACE=(CYL,(10,10))
//VOLPLIST DD DSN=HLQ?.CKZ.WRK.VOLPLIST,
//            DISP=(,CATLG),UNIT=SYSALLDA,
//            RECFM=FB,LRECL=80,BLKSIZE=0,
//            SPACE=(CYL,(1,1))
//CKZIN   DD *
COPY
  FROM-VOLSER(
    SRC001 SRC002
  )
  TO-VOLSER(
    INT001 INT002
  )
  USERCATALOGS(
    SOURCE.USERCAT1 TARGET.USERCAT1
    SOURCE.USERCAT2 TARGET.USERCAT2
  )
  CATWORK-DSN(HLQ?.WRK.* )
  JOURNAL-DDN(JOURNAL)
//*
```

Step 2: Copy INTxxx to TGTxxx without backing up the catalog.

Partial JCL for this step follows:

```
//S1      EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI  DD DISP=SHR,DSN=HLQ?.SCKZPARM(CKZINI)
```

```

//CKZPRINT DD SYSOUT=*
//JOURNAL DD DSN=HLQ?.NUCJRNL,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          RECOG=KS,KEYLEN=64,KEYOFF=0,
//          LRECL=600,SPACE=(CYL,(10,10))
//VOLPLIST DD DSN=HLQ?.CKZNUC.WRK.VOLPLIST,
//           DISP=(,CATLG),UNIT=SYSALLDA,
//           RECFM=FB,LRECL=80,BLKSIZE=0,
//           SPACE=(CYL,(1,1))
//CKZIN DD *
COPY
FROM-VOLSER(
            INT001 INT002
          )
TO-VOLSER(
          TGT001 TGT002
        )
NOUSERCATALOGS
JOURNAL-DDN(JOURNAL)
//*

```

Step 3: Run CKZRNTGT

Partial JCL for this step follows:

```

//* CKZIN IS FROM THE DB2 Cloning Tool COPY COMMAND WITH A
//*   USERCATALOGS KEYWORD, DD VOLPLIST.
//* NUCIN IS FROM THE DB2 Cloning Tool COPY COMMAND WITH THE
//*   NOUSERCATALOGS KEYWORD, DD VOLPLIST.
//* NEWTGT WILL BE USED BY THE DB2 Cloning Tool VOLOPTIIONS COMMAND
//S2 EXEC PGM=IRXJCL,REGION=2M,PARM='CKZRNTGT'
//SYSEXEC DD DSN=HLQ?.SCKZPARAM,DISP=SHR
//SYSTSIN DD DUMMY
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*,DCB=(LRECL=132,RECFM=VBA,BLKSIZE=0)
//CKZIN DD DSN=HLQ?.CKZ.WRK.VOLPLIST,DISP=SHR
//NUCIN DD DSN=HLQ?.CKZNUC.WRK.VOLPLIST,DISP=SHR
//NEWTGT DD DSN=HLQ?.CKZ.WRK.NEWTGT,UNIT=SYSDA,DISP=(,CATLG),
//         DSORG=PS,LRECL=80,RECFM=FB,BLKSIZE=0,
//         SPACE=(CYL,(1,1))

```

Step 4: Run VOLOPTIIONS to update journal for RENAME

Partial JCL for this step follows:

```

//S1 EXEC PGM=CKZ00010,REGION=6M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//JOURNAL DD DISP=SHR,DSN=HLQ?.JRNL
//NEWTGT DD DSN=HLQ?.CKZ.WRK.NEWTGT,UNIT=SYSDA,DISP=SHR
//CKZIN DD *
VOLOPTIIONS UPDATE
NEWTARGETS-DDN(NEWTGT)
JOURNAL-DDN(JOURNAL)
//*

```

Step 5: RENAME data sets on the TGTxxx volumes

Partial JCL and the command for this step follow:

```

//S1 EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//SORTMSG DD SYSOUT=*
//CKZPRINT DD SYSOUT=*

```

```

//DRSTATS DD SYSOUT=*
//JOURNAL DD DSN=HLQ?.JRNL,DISP=SHR
//BCSRECS DD DSN=HLQ?.WRK.BCSRECS,
//          UNIT=SYSALLDA,DISP=(,CATLG),
//          SPACE=(CYL,(10,10))
//VOLBKUP DD DSN=HLQ?.WRK.VOLBKUP,
//          UNIT=SYSALLDA,DISP=(,CATLG),
//          SPACE=(CYL,(40,40))
//CKZIN DD *
RENAME -
SAFE -
VOLBKUP-DDN(VOLBKUP) -
RENAME-MASKS( -
  PROD1.** TEST1.** -
  PROD2.** TEST2.** -
) -
JOURNAL-DDN(JOURNAL)
//*

```

DB2 subsystem cloning from a DB2 BACKUP SYSTEM backup

DB2 subsystems can be cloned using the backup volumes created by a DB2 BACKUP SYSTEM command.

Overview

Because the source DB2 is running at the time of the DB2 BACKUP SYSTEM command, this is an online cloning. This situation is similar to cloning using an interim set of volumes. DB2 Cloning Tool supports this process but requires knowledge of the original source and final target volumes.

If the target volumes are set up in storage groups in a way that mirrors the source volume configuration, it is possible to pair the backup volumes to the target volumes by using the source and target storage group names. For example, if the source has two storage groups, one for logs (SRCSGL) and one for data (SRCSGD), and the target has two storage groups, one for logs (TGTSGL) and one for data (TGTSGD), then the backup volumes can be paired to the target volumes by using the source and target storage group names. The backup volumes that correspond to the source log storage group (SRCSGL) are paired with volumes in the target log storage group (TGTSGL), and the backup volumes that correspond to the source data storage group (SRCSGD) are paired with volumes in the target data storage group (TGTSGD).

The following procedure allows for repetitive cloning without requiring a manual update of the volume specifications when the LAST keyword is used with the DB2GETBACKINFO command.

- The source volumes (referred to as set A) have been copied to the backup volumes (set B) by a DB2 BACKUP SYSTEM command.
- The backup volumes (set B) are copied to the target volumes (set C).
- The data sets on the target volumes (set C) are renamed.
- The target DB2 subsystems on the target volumes are conditioned.
- DB2 Cloning Tool requires the original source volumes (set A) and the final target volumes (set C) to rename and catalog the target volume data sets and to update the target DB2 system to replace the original source volumes with the final target volumes. To accomplish this, DB2 Cloning Tool must know the pairing between set A and set B and then pair those volumes from set A to the final target volumes (set C).

- To obtain the original source volumes (set A) and the backup volumes (set B) from DFSMSHsm, the DB2GETBACKINFO command is executed to query DFSMSHsm. Any generation can be selected in the DB2GETBACKINFO command. In this example, LAST is used for the most current version pairs of set A and set B. This list is provided in DFSMSHsm LIST format in HSMLIST DD. DB2 Cloning Tool also creates the BACKINFO DD data set in a format that is consistent with a utility in DB2 Recovery Expert and Rocket Mainstar DBR for DB2 on z/OS, only for the purpose of keeping cloning from a system backup consistent across products. The HSMLIST DD is not used again. The source ICF catalogs must also be identified by the user in this step.
- The job that executes the BACKINFO-REFORMAT command does several things:
 - Reads the BACKINFO DD data set created in the previous step and creates the VOLPAIRS DD data set with volume pairs set A and set B. These files will be used in subsequent COPY steps.
 - Creates the USERSGDEFS-DDN(USRSGDEF) with the set B volumes and the storage group name of their corresponding source volume, which becomes the input to the clone in the second COPY step.
 - The user catalogs are added to this step and stored in the data set pointed to by UCATS DDN so that they can later be passed to the first COPY command.

For this procedure, two COPY commands are used. Each COPY command must be in its own JCL step.

The first COPY command:

- Uses the DATA-MOVER(PGM(NONE)) keyword and does NOT do any volume copies as the copies (set A to set B) were done by the DB2 BACKUP SYSTEM command.
- Identifies the source volume (set A) to backup volume (set B) pairing by reading the VOLPAIRS-DDN created in the previous step and creates a new data set pointed to by the VOLPLIST DD HLQ?.WRK.VOLPLIST. Another VOLPLIST is created in the second COPY. These will be used later.
- Backs up the source ICF catalog copies from the BACKUP SYSTEM backup volumes.
- Creates a journal data set, which is used throughout the RENAME and DB2 conditioning steps.

The second COPY command:

- Copies the backup volumes (set B) pointed to by FROM-USER-STORAGEGROUP and USERSGDEFS-DDN(USRSGDEF) to the target volumes (set C) pointed to by TO-STORAGEGROUP.
- Volume sets (set B) and (set C) volume pairs are added to a new VOLPLIST data set, HLQ?.NUC.WRK.VOLPLIST.
- The source ICF catalogs are NOT backed up again.
- A journal data set is created but will NOT be used again. The journal from the first copy will be used.

The two VOLPLIST data sets created in each of the copy commands which contain set A to set B and set B to set C (pointed to by DDs CKZIN and NUCIN) are read to create a data set pointed to by DD NEWTGT. This data set contains the pairing set A to set B to set C.

The following step reads the data set pointed to by DD NEWTGT and updates the journal created in the first COPY with the volumes from set A to set C, so the data sets can be renamed and DB2 can be conditioned appropriately.

The RENAME renames and catalogs all of the target volume data sets (set C) to new names.

The DB2 conditioning updates the appropriate DB2 data sets on the target volumes (set C) to the new data set names and volume serials.

If either an active log is defined with more than one stripe or it is a data sharing group, the active logs must be truncated at the point when the DB2 BACKUP SYSTEM FlashCopy of the database volumes completed prior to the first start. This can be accomplished by running the DB2ALTERBSDS command with the SLB-START keyword after DB2UPDATE has run and before DB2START is run. For data sharing, the DB2ALTERBSDS command with the SLB-START keyword should be run for each member of the target data sharing group.

For data sharing, the first members started will receive a DSNR020I WTOR due to the conditional restart records contained in the other members. If the start jobs are being run serially by a job scheduler, it is recommended that the DB2START command for all but the last member include the STOP-WAITING-IF-DSNR020I keyword with a value of Y. This will allow the starts of the other members to be done concurrently. After the last member has been started, a DB2START for the first member should be done again using the WAITONLY and DB2-ALREADY-RUNNING(RC(0)) keywords. Completion of the DB2START with a return code of 4 or less indicates that the first member is up and ready for the remainder of the DB2 conditioning jobs.

Step overview

This section summarizes the steps you need to follow to perform DB2 subsystem cloning.

1. Run DB2GETBACKINFO to get the source (set A) and backup (set B) volume pairing used by the DB2 BACKUP SYSTEM command. (Any available generation can be specified; our example uses LAST.)
2. Run BACKINFO-REFORMAT to take the backinfo data set created by the DB2GETBACKINFO command (Step 1) and reformat it for use by subsequent COPY commands (Steps 3 and 4).
3. Run COPY with DATA-MOVER(PGM(NONE)). Use VOLPAIRS-DDN to get the source volumes (set A) to backup volumes (set B) pairing (data set from step 2), use a USERCATALOGS-DDN keyword (data set from step 2), and include the VOLPLIST DD.

The DD name is VOLPLIST and it must have attributes of RECFM=FB,LRECL=80. The data set allocated by the VOLPLIST DD will be used by a subsequent step/job.

4. Run COPY using FROM-USER-STORAGEGROUP and USERSGDEFS-DDN to get the backup volumes (set B) (data set from step 2), using the NOUSERCATALOGS keyword and including the VOLPLIST DD.

The DD name is VOLPLIST and it must have attributes of RECFM=FB,LRECL=80. The data set allocated by the VOLPLIST DD will be used by a subsequent step/job.

5. Run CKZRNTGT with the VOLPLIST from step 3 on the CKZIN DD and the VOLPLIST from step 4 on the NUCIN DD.

6. Run VOLOPTIONS using NEWTGT data set as input to NEWTARGETS-DDN.
7. Run RENAME for target volumes (set C).
8. Run the DB2 conditioning commands: DB2UPDATE, DB2START, DB2FIX DATABASES(DB2), DB2SQL, DB2FIX DATABASES(APPLICATION), DB2STOP, DB2UTILXCLEAN.

Note: Steps 3, 6, 7, and 8 use the same journal data set. Step 4 uses a different journal data set.

Example

In the example steps that follow, the following items are used:

- The source DB2 resides on the source volumes (SRCxxx). The source log volumes are in storage group SRCSTGL and the source data volumes are in storage group SRCSTGD.
- The target DB2 will reside on the target volumes (TGTxxx). The target log volumes are in storage group TGTSTGL and the target data volumes are in storage group TGTSTGD.
- There are one or more sets of backup volumes (BKPxxx) that have been created by DB2 BACKUP SYSTEM commands.
- There are two source ICF catalogs (USERCAT.SRC01 and USERCAT.SRC02) that reside on source volumes.

Step 1

The first step is to get the source to backup volume pairing (SRCxxx to BKPxxx) and identify the source ICF catalogs from the last DB2 BACKUP SYSTEM taken for location DB2PLOC. Sample JCL can be found in the installation SCKZJCL library in member CKZDGETB.

Partial JCL for this step follows:

```
//S1      EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DSN=HLQ?.SCKZLOAD,DISP=SHR
//CKZINI  DD DSN=HLQ?.SCKZPARAM(CKZINI),DISP=SHR
//CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//BACKINFO DD DSN=HLQ?.WRK.BACKINFO,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          SPACE=(CYL,(1,1))
//HSMLIST DD DSN=HLQ?.WRK.HSMLIST,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          SPACE=(CYL,(1,1))
//CKZIN   DD *
          DB2GETBACKINFO           -
          BACKINFO-DDN(BACKINFO)   -
          WORK-DDN(HSMLIST)        -
          LAST                     -
          LOCATION(DB2PLOC)        -
          USERCATALOGS(            -
            USERCAT.SRC01          -
            USERCAT.SRC02          -
          )
//*
```

Step 2

The second step is to reformat the output of step 1 (backinfo data set) for use in the COPY in steps 3 and 4. The user catalog pairs are also specified here. Sample JCL can be found in the installation SCKZJCL library in member CKZBKIRF.

Partial JCL for this step follows:

```
//S1      EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DSN=HLQ?.SCKZLOAD,DISP=SHR
//CKZINI  DD DSN=HLQ?.SCKZPARAM(CKZINI),DISP=SHR
//CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//BACKINFO DD DISP=SHR,DSN=HLQ?.WRK.BACKINFO
//VOLPAIRS DD DSN=HLQ?.WRK.VOLPAIRS,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          SPACE=(CYL,(1,1))
//USRSGDEF DD DSN=HLQ?.WRK.USRSGDEF,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          SPACE=(CYL,(1,1))
//UCATS   DD DSN=HLQ?.WRK.UCATS,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          SPACE=(CYL,(1,1))
//CKZIN   DD *
        BACKINFO-REFORMAT          -
        BACKINFO-DDN(BACKINFO)     -
        VOLPAIRS-DDN(VOLPAIRS)     -
        USRSGDEF-DDN(USRSGDEF)     -
        USERCATALOGS-DDN(UCATS)    -
        USERCATALOGS(
            USERCAT.SRC01 USERCAT.TGT01 -
            USERCAT.SRC02 USERCAT.TGT02 -
        )
//*
```

Step 3

The third step is to set the pairing between source (SRCxxx) and backup (BKPxxx) volumes in the journal and back up the source ICF catalogs from the backup volumes. Sample JCL can be found in the installation SCKZJCL library in member CKZCOPY.

Partial JCL for this step follows:

```
//S1      EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI  DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//VOLPAIRS DD DISP=SHR,DSN=HLQ?.WRK.VOLPAIRS
//UCATS   DD DISP=SHR,DSN=HLQ?.WRK.UCATS
//JOURNAL DD DSN=HLQ?.JRNL,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          RECOG=KS,KEYLEN=64,KEYOFF=0,
//          LRECL=600,SPACE=(CYL,(10,10))
//VOLPLIST DD DSN=HLQ?.WRK.VOLPLIST,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          RECFM=FB,LRECL=80,BLKSIZE=0,
//          SPACE=(CYL,(1,1))
//CKZIN   DD *
        COPY                          -
        DATA-MOVER(PGM(NONE))        -
        VOLPAIRS-DDN(VOLPAIRS)       -
```

```

        USERCATALOGS-DDN(UCATS) -
        CATWORK-DSN(HLQ?.WRK.* ) -
        JOURNAL-DDN(JOURNAL)
//*
```

Step 4

The fourth step is to copy the backup volumes (BKPxxx) to the target volumes (TGTxxx) without backing up any ICF catalogs. Sample JCL can be found in the installation SCKZJCL library in member CKZCOPY.

Partial JCL for this step follows:

```

//S1      EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI  DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//USRSGDEF DD DISP=SHR,DSN=HLQ?.WRK.USRSGDEF
//JOURNAL DD DSN=HLQ?.NUCJRNL,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          RECOG=KS,KEYLEN=64,KEYOFF=0,
//          LRECL=600,SPACE=(CYL,(10,10))
//VOLPLIST DD DSN=HLQ?.NUC.WRK.VOLPLIST,
//            DISP=(,CATLG),UNIT=SYSALLDA,
//            RECFM=FB,LRECL=80,BLKSIZE=0,
//            SPACE=(CYL,(1,1))
//CKZIN   DD *
COPY
FROM-USER-STORAGEGROUP( -
                        SRCSSL SRCSGD -
                        ) -
USRSGDEFS-DDN(USRSGDEF) -
USRSGDEFS-OFFSETS(VOLSER(1) SGNAME(8) INCLEXCL(18)) -
TO-STORAGEGROUP( -
                 TGTSGL TGTSGD -
                 ) -
NOUSERCATALOGS -
JOURNAL-DDN(JOURNAL)
//*
```

Step 5

The fifth step is to run CKZRNTGT with the VOLPLIST data sets from steps 3 and 4. Sample JCL can be found in the installation SCKZJCL library in member CKZRNTGT.

Partial JCL for this step follows:

```

//* CKZIN IS FROM THE DB2 Cloning Tool COPY COMMAND WITH A
//* USERCATALOGS KEYWORD, DD VOLPLIST.
//* NUCIN IS FROM THE DB2 Cloning Tool COPY COMMAND WITH THE
//* NOUSERCATALOGS KEYWORD, DD VOLPLIST.
//* NEWTGT WILL BE USED BY THE DB2 Cloning Tool VOLOPTIIONS COMMAND
//S2      EXEC PGM=IRXJCL,REGION=2M,PARM='CKZRNTGT'
//SYSEXEC DD DSN=HLQ?.SCKZPARAM,DISP=SHR
//SYSTSIN DD DUMMY
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*,DCB=(LRECL=132,RECFM=VBA,BLKSIZE=0)
//CKZIN   DD DSN=HLQ?.WRK.VOLPLIST,DISP=SHR
//NUCIN   DD DSN=HLQ?.NUC.WRK.VOLPLIST,DISP=SHR
//NEWTGT  DD DSN=HLQ?.WRK.NEWTGT,UNIT=SYSDA,DISP=(,CATLG),
//          DSORG=PS,LRECL=80,RECFM=FB,BLKSIZE=0,
//          SPACE=(CYL,(1,1))
```

Step 6

This is the sixth step of run VOLOPTIONS with the newtgt data set from step 5 to update the journal with the actual source (SRCxxx) to target (TGTxxx) volume pairing for RENAME. Sample JCL can be found in the installation SCKZJCL library in member CKZVOLOP.

Partial JCL for this step follows:

```
//S1 EXEC PGM=CKZ00010,REGION=6M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//JOURNAL DD DISP=SHR,DSN=HLQ?.JRNL
//NEWTGT DD DISP=SHR,DSN=HLQ?.WRK.NEWTGT
//CKZIN DD *
        VOLOPTIONS UPDATE -
        NEWTARGETS-DDN(NEWTGT) -
        JOURNAL-DDN(JOURNAL)
//*
```

Step 7

The seventh step of the RENAME the data sets on the target volumes (TGTxxx). Sample JCL can be found in the installation SCKZJCL library in member CKZREN.

Partial JCL and the RENAME command for this step follow:

```
//S1 EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//SORTMSG DD SYSOUT=*
//CKZPRINT DD SYSOUT=*
//DRSTATS DD SYSOUT=*
//JOURNAL DD DSN=HLQ?.JRNL,DISP=SHR
//BCSRECS DD DSN=HLQ?.WRK.BCSRECS,
// UNIT=SYSALLDA,DISP=(,CATLG),
// SPACE=(CYL,(10,10))
//VOLBKUP DD DSN=HLQ?.WRK.VOLBKUP,
// UNIT=SYSALLDA,DISP=(,CATLG),
// SPACE=(CYL,(40,40))
//CKZIN DD *
        RENAME -
        SAFE -
        VOLBKUP-DDN(VOLBKUP) -
        RENAME-MASKS( -
                PROD1.** TEST1.** -
                PROD2.** TEST2.** -
                ) -
        JOURNAL-DDN(JOURNAL)
//*
```

Step 8

The last step is to run the DB2 conditioning commands. These commands are:

- DB2UPDATE
- DB2ALTERBSDS SLB-START
- DB2START SPECIAL
- DB2FIX DATABASES(DB2)
- DB2SQL
- DB2FIX DATABASES(APPLICATION)

- DB2STOP
- DB2UTILXCLEAN

For additional information about the conditioning commands, refer to the appropriate section in “DB2 online cloning” on page 104 for your specific type of cloning.

If either an active log is defined with more than one stripe or it is a data sharing group, the active logs must be truncated at the point when the DB2 BACKUP SYSTEM FlashCopy of the database volumes completed prior to the first start. This can be accomplished by running the DB2ALTERBSDS command with the SLB-START keyword after DB2UPDATE has run and before DB2START is run. For data sharing, the DB2ALTERBSDS command with the SLB-START keyword should be run for each member of the target data sharing group.

The following are examples of the DB2UPDATE, DB2ALTERBSDS, and DB2START steps necessary to handle this condition.

Example 1 – a non data sharing subsystem with active logs that are striped:

1. Update the DB2 directory and the BSDS.

```
DB2UPDATE                -
  DB2-HLQS( srcvcat tgtvcat ) -
  DDF( ... )              -
  JOURNAL-DDN(JOURNAL)
```

2. Create a system-level backup (SLB) start conditional restart record in the BSDS.

```
DB2ALTERBSDS            -
  SLB-START              -
  JOURNAL-DDN(JOURNAL)
```

3. Start the target DB2 in special mode and automatically reply to the DB2 restart WTOR.

```
DB2START                 -
  DB2-SSID(ssid)         -
  SPECIAL                 -
  DSNZPARM(ssidSPEC)     -
  REPLY-TO-RESTART-WTOR(Y)
```

Example 2 – a data sharing group with three members (the active logs may or may not be striped):

1. Update the DB2 directory and the BSDS for mbr1.

```
DB2UPDATE                -
  DB2-HLQS( srcvcat tgtvcat ) -
  DB2-GROUP( srcgrp tgtgrp ) -
  DB2-MEMBERS( srcmbr1 mbr1   -
               srcmbr2 mbr2   -
               srcmbr3 mbr3 ) -
  DDF( ... )              -
  JOURNAL-DDN(JOURNAL)
```

2. Update the BSDS for mbr2.

```
DB2UPDATE                -
  BSDSONLY                -
  DB2-HLQS( srcvcat tgtvcat ) -
  DB2-GROUP( srcgrp tgtgrp ) -
  DB2-MEMBERS( srcmbr1 mbr1   -
               srcmbr2 mbr2   -
               srcmbr3 mbr3 ) -
  DDF( ... )              -
  JOURNAL-DDN(JOURNAL)
```

3. Update the BSDS for mbr3.

```

DB2UPDATE                -
  BSDSONLY                -
  DB2-HLQS( srcvcat tgtvcat ) -
  DB2-GROUP( srcgrp tgtgrp ) -
  DB2-MEMBERS( srcmbr1 mbr1
    srcmbr2 mbr2
    srcmbr3 mbr3 )         -
  DDF( ... )              -
  JOURNAL-DDN(JOURNAL)

```

4. Create a system-level backup (SLB) start conditional restart record in the BSDS for mbr1.

```

DB2ALTERBSDS            -
  DB2-MEMBER(mbr1)      -
  SLB-START              -
  JOURNAL-DDN(JOURNAL)

```

5. Create a system-level backup (SLB) start conditional restart record in the BSDS for mbr2.

```

DB2ALTERBSDS            -
  DB2-MEMBER(mbr2)      -
  SLB-START              -
  JOURNAL-DDN(JOURNAL)

```

6. Create a system-level backup (SLB) start conditional restart record in the BSDS for mbr3.

```

DB2ALTERBSDS            -
  DB2-MEMBER(mbr3)      -
  SLB-START              -
  JOURNAL-DDN(JOURNAL)

```

7. Start the target DB2 mbr1 in special mode, automatically reply to the DB2 restart WTOR, and stop waiting if a DSNR020I WTOR is received.

```

DB2START                -
  DB2-SSID(mbr1)        -
  SPECIAL                -
  DSNZPARM(mbr1SPEC)    -
  REPLY-TO-RESTART-WTOR(Y) -
  STOP-WAITING-IF-DSNR020I(Y)

```

8. Start the target DB2 mbr2 in special and light mode, automatically reply to the DB2 restart WTOR, and stop waiting if a DSNR020I WTOR is received.

```

DB2START                -
  DB2-SSID(mbr2)        -
  SPECIAL                -
  DSNZPARM(mbr2SPEC)    -
  LIGHT                  -
  REPLY-TO-RESTART-WTOR(Y) -
  STOP-WAITING-IF-DSNR020I(Y)

```

9. Start the target DB2 mbr3 in special and light mode and automatically reply to the DB2 restart WTOR.

```

DB2START                -
  DB2-SSID(mbr3)        -
  SPECIAL                -
  DSNZPARM(mbr3SPEC)    -
  LIGHT                  -
  REPLY-TO-RESTART-WTOR(Y)

```

10. Wait for the target DB2 mbr1 to complete its startup.

```

DB2START                -
  DB2-SSID(mbr1)        -
  SPECIAL                -
  DSNZPARM(mbr1SPEC)    -
  REPLY-TO-RESTART-WTOR(Y) -
  DB2-ALREADY-RUNNING(RC(0)) -
  WAITONLY

```

DB2 subsystem cloning to a specific point in time from a DB2 BACKUP SYSTEM backup

DB2 subsystems can be cloned by using the backup volumes that are created by a DB2 BACKUP SYSTEM command. As of DB2 V11, it is possible to clone to a specific point in time that is different from the point in time when the BACKUP SYSTEM was taken. This cloning scenario uses the RESTORE SYSTEM utility run using LOGONLY SWITCH VCAT SYSVALUEDDN(*ddname*). The point in time to use is set by using the DSNJU003 (change log inventory) utility to create a conditional restart record using SYSPITR or SYSPITRT.

Overview

Because the source DB2 is running at the time of the DB2 BACKUP SYSTEM command, this is an online cloning. This situation is similar to cloning by using an interim set of volumes. DB2 Cloning Tool supports this process, but requires knowledge of the original source and final target volumes.

For this scenario, there must be at least two DB2 system level backups (SLBs) for the DB2 system that is being cloned. The cloning will be to a specific point in time that lies between two of the SLBs. For example, there are SLBs at points-in-time 1 and 2, and the cloning will be to point-in-time 1.5. The cloning uses the data volumes from point-in-time 1 and the log volumes from point-in-time 2. The RESTORE SYSTEM utility is used to apply the logs up to the required point in time 1.5.

This scenario references volume sets in the following manner:

Set A The source volumes.

Set B The system level backup volume. Subsets of these volumes are:

- Set B1 is the SLB volume set for point-in-time 1.
- Set B2 is the SLB volume set for point-in-time 2.

Set C Target volumes.

Data (D) or Log (L) volumes

To specifically identify the data and log volumes from each set, a D (for data volumes) or an L (for log volumes) is appended the volume set name. For example: B1D is the backup data volume set for point-in-time 1.

- The source volumes (set A) already have been copied to the backup volumes (sets B1 and B2) by DB2 BACKUP SYSTEM commands.
- The backup data volumes for point-in-time 1 (set B1D) are copied to the target data volumes (set CD) and the backup log volumes for point-in-time 2 (set B2L) are copied to the target log volumes (set CL).
- The data sets on the target volumes (set C) are renamed.
- The target DB2 subsystems BSDS are conditioned.
- The conditional restart records using SYSPITR or SYSPITRT are created.
- The RESTORE SYSTEM utility is run with LOGONLY SWITCH VCAT SYSVALUEDDN(*ddname*).
- The target DB2 subsystems on the target volumes are conditioned.
- DB2 Cloning Tool requires the original source volumes (set A) and the final target volumes (set C) to rename and catalog the target volume data sets, and to update the target DB2 system to replace the original source volumes with the

final target volumes. To accomplish this, DB2 Cloning Tool must know the pairing between set A and set B1 and then pair those volumes from set A to the final target volumes (set C).

- To obtain the original source volumes (set A) and the backup volumes for point-in-time 1 (set B1) and point-in-time 2 (set B2) from DFSMSHsm, the DB2GETBACKINFO command is executed to query DFSMSHsm. Any generation can be selected in the DB2GETBACKINFO command. In this example, the token that applies to the point-in-time 1 SLB is used to get the list of data volume pairs of set A and set B1, and the token that applies to the point-in-time 2 SLB is used to get the list of log volume pairs of set A and set B2. The lists are provided in DFSMSHsm LIST format in HSMLIST DD. DB2 Cloning Tool also creates the BACKINFO DD data set in a format that is consistent with a utility in DB2 Recovery Expert and Rocket Mainstar DBR for DB2 on z/OS, only for the purpose of keeping cloning from a system backup consistent across products. The HSMLIST DD is not used again. The source ICF catalogs must also be identified by the user in this step.
- The job that executes the BACKINFO-REFORMAT command:
 - Reads the BACKINFO DD data sets created in the previous step and creates the VOLPAIRS DD data set with volume pairs set A and set B. These files are used in subsequent COPY steps.
 - Creates the USERSGDEFS-DDN(USRSGDEF) with the set B volumes and the storage group name of their corresponding source volume, which becomes the input to the clone in the second COPY step.
 - The user catalogs are added to this step and stored in the data set pointed to by UCATS DDN so that they can later be passed to the first COPY command.

For this part of the procedure, two COPY commands are used. Each COPY command must be in its own JCL step.

The first COPY command:

- Uses the DATA-MOVER(PGM(NONE)) keyword and does NOT do any volume copies, as the copies (set A to set B) were done by the DB2 BACKUP SYSTEM commands.
- Identifies the source volume (set A) to backup volume (set B) pairing by reading the VOLPAIRS-DDN created in the previous step and creates a new data set pointed to by the VOLPLIST DD HLQ?.WRK.VOLPLIST. Another VOLPLIST is created in the second COPY. These are used later.
- Backs up the source ICF catalog copies from the BACKUP SYSTEM backup volumes.
- Creates a journal data set, which is the journal data set that is used throughout the RENAME and DB2 conditioning steps.

The second COPY command:

- Copies the backup volumes (set B) contained in the DD pointed to by the FROM-USER-STORAGEGROUP and USERSGDEFS-DDN(USRSGDEF) to the target volumes (set C) pointed to by the TO-STORAGEGROUP.
- Volume sets (set B) and (set C) volume pairs are added to a new VOLPLIST data set, HLQ?.NUC.WRK.VOLPLIST.
- The source ICF catalogs are NOT backed up again.
- A journal data set is created, but will not be used again. The journal from the first copy is used.

The two VOLPLIST data sets created in each of the COPY commands, which contain set A to set B and set B to set C (pointed to by DDs CKZIN and NUCIN), are read to create a data set pointed to by DD NEWTGT. This data set contains the pairing set A to set B to set C.

Next, the data set pointed to by DD NEWTGT is read and the journal that was created in the first COPY command is updated with the volumes from set A to set C, so the data sets can be renamed and DB2 can be conditioned appropriately.

The RENAME renames and catalogs all of the target volume data sets (set C) to new names.

The DB2 BSDS conditioning updates the DB2 BSDS and directory data sets on the target volumes (set C) to the new data set names and volume serials and creates the SYSVALUE data set that will be used by the RESTORE SYSTEM utility.

The conditional restart records are created for the required point in time using SYSPITR or SYSPITRT.

The RESTORE SYSTEM utility is run using LOGONLY SWITCH VCAT SYSVALUEDDN(ddname). This will bring the DB2 system to the required point in time.

DB2 Cloning Tool updates to the DB2 directory and catalog may have been regressed by the RESTORE SYSTEM, so the DB2 conditioning is done again.

Step overview

This section summarizes the steps you need to follow to perform DB2 subsystem cloning from the SLB to a specific point in time.

1. Run DB2GETBACKINFO to get the source data (set AD) and backup (set B1D) volume pairing used by the DB2 BACKUP SYSTEM command. (Any available generation can be specified; this example uses a token for the SLB of point in time 1.)
2. Run DB2GETBACKINFO to get the source log (set AL) and backup (set B2L) volume pairing used by the DB2 BACKUP SYSTEM command. (The generation specified here should be the one that follows the generation used in step 1; this example uses a token for the SLB of point in time 2.)
3. Run BACKINFO-REFORMAT to take the backinfo data sets created by the DB2GETBACKINFO commands (Steps 1 and 2) and reformat it for use by subsequent COPY commands (Steps 4 and 5).
4. Run COPY with DATA-MOVER(PGM(NONE)). Use VOLPAIRS-DDN to get the source volumes (set A) to backup volumes (set B) pairing (data set from step 3), use a USERCATALOGS-DDN keyword (data set from step 3), and include the VOLPLIST DD. The DD name is VOLPLIST and it must have attributes of RECFM=FB,LRECL=80. The data set allocated by the VOLPLIST DD will be used by a subsequent step/job.
5. Run COPY using FROM-USER-STORAGEGROUP and USERSGDEFS-DDN to get the backup volumes (set B) (data set from step 3), using the NOUSERCATALOGS keyword and including the VOLPLIST DD. The DD name is VOLPLIST and it must have attributes of RECFM=FB,LRECL=80. The data set allocated by the VOLPLIST DD will be used by a subsequent step/job.

6. Run CKZRNTGT with the VOLPLIST from step 4 on the CKZIN DD and the VOLPLIST from step 5 on the NUCIN DD.
7. Run VOLOPTIONS using NEWTGT data set as input to NEWTARGETS-DDN.
8. Run RENAME for target volumes (set C).
9. Run DB2UPDATE to condition the BSDS and directory.
10. Create the conditional restart record for required point in time.
11. Start the DB2 systems.
12. Run the RESTORE SYSTEM utility using LOGONLY SWITCH VCAT SYSVALUEDDN(*ddname*).
13. Stop the DB2 systems.
14. Run the DB2 conditioning commands: DB2UPDATE, DB2START, DB2FIX DATABASES(DB2), DB2SQL, DB2FIX DATABASES(APPLICATION), DB2STOP, DB2UTILXCLEAN.

Note: Steps 4, 7, 8, 9, and 14 use the same journal data set. Step 5 uses a different journal data set.

Example

In the example steps that follow, the following items are assumed:

- The source DB2 resides on the source volumes (SRCxxx). The source log volumes are in storage group SRCISGL and the source data volumes are in storage group SRCISGD.
- The target DB2 will reside on the target volumes (TGTxxx). The target log volumes are in storage group TGTISGL and the target data volumes are in storage group TGTISGD.
- There are at least two sets of backup volumes (BKPxxx) that have been created by DB2 BACKUP SYSTEM commands.
- There are two source ICF catalogs (USERCAT.SRCLG and USERCAT.SRCDB) that reside on source volumes. The USERCAT.SRCLG catalog is on a source log volume and contains entries for the data sets on the source log volumes. The USERCAT.SRCDB catalog is on a source data volume and contains entries for the data sets on the source data volumes.
- There are two target catalogs (USERCAT.TGTISGL and USERCAT.TGTISGD) that do not reside on target volumes. The USERCAT.TGTISGL catalog contains entries for the data sets on the target log volumes. The USERCAT.SRCDB catalog contains entries for the data sets on the target data volumes.
- There are two DB2 system level backups:
 - A backup for a point in time at 2013/07/06 11:10:15 that has a token of X'C5C2F1C1CB9E97E16641BD66000000000027217E83A'
 - A backup for a point in time at 2013/07/19 10:59:19 that has a token of X'C5C2F1C1CBAEEDAF121378A000000000002ADD6A3F0'
- The intention is to create the target DB2 at a point in time of 2013/07/17 12:00:00.

Step 1

The first step is to get the source to backup data volume pairing (SRCxxx to BKPxxx) and identify the source ICF catalogs from the last DB2 BACKUP SYSTEM taken for location DB2PLOC for a point in time at 2013/07/06 11:10:15. Sample JCL can be found in the installation library SCKZJCL in member CKZDGETB.

Partial JCL for this step follows:

```
//S1      EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DSN=HLQ?.SCKZLOAD,DISP=SHR
//CKZINI  DD DSN=HLQ?.SCKZPARM(CKZINI),DISP=SHR
//CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//BACKINFO DD DSN=HLQ?.WRK.BACKINFD,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          SPACE=(CYL,(1,1))
//HSMLIST DD DSN=HLQ?.WRK.HSMLIST,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          SPACE=(CYL,(1,1))
//CKZIN   DD *
          DB2GETBACKINFO -
            DATABASESONLY -
            BACKINFO-DDN(BACKINFO) -
            WORK-DDN(HSMLIST) -
            TOKEN(X'C5C2F1C1CB9E97E16641BD6600000000027217E83A') -
            LOCATION(DB2PLOC) -
            USERCATALOGS( -
              USERCAT.SRCDB -
            )
//*
```

Step 2

This step retrieves the source to backup log volume pairing (SRCxxx to BKPxxx) and identifies the source ICF catalogs from the last DB2 BACKUP SYSTEM taken for location DB2PLOC for a point in time at 2013/07/19 10:59:19. Sample JCL can be found in the installation library SCKZJCL in member CKZDGETB.

Partial JCL for this step follows:

```
//S1      EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DSN=HLQ?.SCKZLOAD,DISP=SHR
//CKZINI  DD DSN=HLQ?.SCKZPARM(CKZINI),DISP=SHR
//CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//BACKINFO DD DSN=HLQ?.WRK.BACKINFL,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          SPACE=(CYL,(1,1))
//HSMLIST DD DSN=HLQ?.WRK.HSMLIST,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          SPACE=(CYL,(1,1))
//CKZIN   DD *
          DB2GETBACKINFO -
            LOGSONLY -
            BACKINFO-DDN(BACKINFO) -
            WORK-DDN(HSMLIST) -
            TOKEN(X'C5C2F1C1CBAEEDAF121378A00000000002ADD6A3F0') -
            LOCATION(DB2PLOC) -
            USERCATALOGS( -
              USERCAT.SRCLG -
            )
//*
```

Step 3

This step reformats the outputs of steps 1 and 2 (backinfo data sets) for use in the COPY in steps 4 and 5. The user catalog pairs are also specified here. Sample JCL can be found in the installation library SCKZJCL in member CKZBKIRF.

Partial JCL for this step follows:

```

//S1      EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DSN=HLQ?.SCKZLOAD,DISP=SHR
//CKZINI  DD DSN=HLQ?.SCKZPARAM(CKZINI),DISP=SHR
//CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//BACKINFO DD DISP=SHR,DSN=HLQ?.WRK.BACKINFD
//        DD DISP=SHR,DSN=HLQ?.WRK.BACKINFL
//VOLPAIRS DD DSN=HLQ?.WRK.VOLPAIRS,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          SPACE=(CYL,(1,1))
//USRSGDEF DD DSN=HLQ?.WRK.USRSGDEF,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          SPACE=(CYL,(1,1))
//UCATS   DD DSN=HLQ?.WRK.UCATS,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          SPACE=(CYL,(1,1))
//CKZIN   DD *
        BACKINFO-REFORMAT -
        BACKINFO-DDN(BACKINFO) -
        VOLPAIRS-DDN(VOLPAIRS) -
        USRSGDEF-DDN(USRSGDEF) -
        USERCATALOGS-DDN(UCATS) -
        USERCATALOGS( -
            USERCAT.SRCLG USERCAT.TGTLG -
            USERCAT.SRCDB USERCAT.TGTDB -
        )
//*

```

Step 4

This step sets the pairing between source (SRCxxx) and backup (BKPxxx) volumes in the journal, and backs up the source ICF catalogs from the backup volumes. Sample JCL can be found in the installation library SCKZJCL in member CKZCOPY.

Partial JCL for this step follows:

```

//S1      EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI  DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//VOLPAIRS DD DISP=SHR,DSN=HLQ?.WRK.VOLPAIRS
//UCATS   DD DISP=SHR,DSN=HLQ?.WRK.UCATS
//JOURNAL DD DSN=HLQ?.JRNL,
//        DISP=(,CATLG),UNIT=SYSALLDA,
//        RECOFG=KS,KEYLEN=64,KEYOFF=0,
//        LRECL=600,SPACE=(CYL,(10,10))
//VOLPLIST DD DSN=HLQ?.WRK.VOLPLIST,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          RECFM=FB,LRECL=80,BLKSIZE=0,
//          SPACE=(CYL,(1,1))
//CKZIN   DD *
        COPY -
        DATA-MOVER(PGM(NONE)) -
        VOLPAIRS-DDN(VOLPAIRS) -
        USERCATALOGS-DDN(UCATS) -
        CATWORK-DSN(HLQ?.WRK.* ) -
        JOURNAL-DDN(JOURNAL)
//*

```

Step 5

This step copies the backup volumes (BKPxxx) to the target volumes (TGTxxx) without backing up any ICF catalogs. Sample JCL can be found in the installation library SCKZJCL in member CKZCOPY.

Partial JCL for this step follows:

```
//S1 EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//USRSGDEF DD DISP=SHR,DSN=HLQ?.WRK.USRSGDEF
//JOURNAL DD DSN=HLQ?.NUCJRNL,
// DISP=(,CATLG),UNIT=SYSALLDA,
// RECOG=KS,KEYLEN=64,KEYOFF=0,
// LRECL=600,SPACE=(CYL,(10,10))
//VOLPLIST DD DSN=HLQ?.NUC.WRK.VOLPLIST,
// DISP=(,CATLG),UNIT=SYSALLDA,
// RECFM=FB,LRECL=80,BLKSIZE=0,
// SPACE=(CYL,(1,1))
//CKZIN DD *
COPY -
  FROM-USER-STORAGEGROUP( -
    SRCSGL SRCSGD -
  ) -
  TO-STORAGEGROUP( -
    TGTSGL TGTSGD -
  ) -
  USRSGDEFS-DDN(USRSGDEF) -
  USRSGDEFS-OFFSETS(VOLSER(1) SGNAME(8) INCLEXCL(18)) -
  NOUSERCATALOGS -
  JOURNAL-DDN(JOURNAL)
/*
```

Step 6

This step runs CKZRNTGT with the VOLPLIST data sets from steps 4 and 5. Sample JCL can be found in the installation library SCKZJCL in member CKZRNTGT.

Partial JCL for this step follows:

```
/* CKZIN IS FROM THE DB2 Cloning Tool COPY COMMAND WITH A
/* USERCATALOGS KEYWORD, DD VOLPLIST.
/* NUCIN IS FROM THE DB2 Cloning Tool COPY COMMAND WITH THE
/* NOUSERCATALOGS KEYWORD, DD VOLPLIST.
/* NEWTGT WILL BE USED BY THE DB2 Cloning Tool VOLOPTIIONS COMMAND
//S2 EXEC PGM=IRXJCL,REGION=2M,PARM='CKZRNTGT'
//SYSEXEC DD DSN=HLQ?.SCKZPARAM,DISP=SHR
//SYSTSIN DD DUMMY
//SYSTSPRT DD SYSOUT=*
//SYSSPRINT DD SYSOUT=*,DCB=(LRECL=132,RECFM=VBA,BLKSIZE=0)
//CKZIN DD DSN=HLQ?.WRK.VOLPLIST,DISP=SHR
//NUCIN DD DSN=HLQ?.NUC.WRK.VOLPLIST,DISP=SHR
//NEWTGT DD DSN=HLQ?.WRK.NEWTGT,UNIT=SYSDA,DISP=(,CATLG),
// DSORG=PS,LRECL=80,RECFM=FB,BLKSIZE=0,
// SPACE=(CYL,(1,1))
```

Step 7

This step runs VOLOPTIIONS with the NEWTGT data set from step 6 to update the journal with the actual source (SRCxxx) to target (TGTxxx) volume pairing for RENAME. Sample JCL can be found in the installation library SCKZJCL in member CKZVOLOP.

Partial JCL for this step follows:

```
//S1 EXEC PGM=CKZ00010,REGION=6M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
```

```

| //CKZPRINT DD SYSOUT=*
| //SYSUDUMP DD SYSOUT=*
| //JOURNAL DD DISP=SHR,DSN=HLQ?.JRNL
| //NEWTGT DD DISP=SHR,DSN=HLQ?.WRK.NEWGTG
| //CKZIN DD *
|     VOLOPTIONS UPDATE -
|     NEWTARGETS-DDN(NEWTGT) -
|     JOURNAL-DDN(JOURNAL)
| //*

```

Step 8

This step RENAMES the data sets on the target volumes (TGTxxx). Sample JCL can be found in the installation library SCKZJCL in member CKZREN.

Partial JCL and the RENAME command for this step follow:

```

| //S1 EXEC PGM=CKZ00010,REGION=8M
| //STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
| //CKZINI DD DISP=SHR,DSN=HLQ?.SCKZPARM(CKZINI)
| //SORTMSG DD SYSOUT=*
| //CKZPRINT DD SYSOUT=*
| //DRSTATS DD SYSOUT=*
| //JOURNAL DD DSN=HLQ?.JRNL,DISP=SHR
| //BCSRECS DD DSN=HLQ?.WRK.BCSRECS,
| // UNIT=SYSALLDA,DISP=(,CATLG),
| // SPACE=(CYL,(10,10))
| //VOLBKUP DD DSN=HLQ?.WRK.VOLBKUP,
| // UNIT=SYSALLDA,DISP=(,CATLG),
| // SPACE=(CYL,(40,40))
| //CKZIN DD *
|     RENAME -
|     SAFE -
|     VOLBKUP-DDN(VOLBKUP) -
|     RENAME-MASKS( -
|     PROD1.** TEST1.** -
|     PROD2.** TEST2.** -
|     ) -
|     JOURNAL-DDN(JOURNAL)
| //*

```

Step 9

This step conditions the BSDS and DB2 directory, and creates the SYSVALUE data for use by the later RESTORE SYSTEM step. This example is for a non-data sharing subsystem. Sample JCL can be found in the installation library SCKZJCL in member CKZDUPD.

Partial JCL for this step follows:

```

| //S1 EXEC PGM=CKZ00010,REGION=6M
| //STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
| //CKZINI DD DISP=SHR,DSN=HLQ?.SCKZPARM(CKZINI)
| //CKZPRINT DD SYSOUT=*
| //SYSVALUE DD DSN=HLQ?.WRK.SYSVALUE,
| // DISP=(,CATLG),UNIT=SYSALLDA,
| // RECFM=FB,LRECL=80,BLKSIZE=0,
| // SPACE=(CYL,(1,1))
| //JOURNAL DD DISP=SHR,DSN=HLQ?.JRNL
| //BSDS01 DD DISP=OLD,DSN=TEST1.BSDS01
| //BSDS02 DD DISP=OLD,DSN=TEST1.BSDS02
| //SYSDBDXA DD DISP=OLD,
| // DSN=TEST1.DSNDBC.DSNDB01.SYSDBDXA.I0001.A001
| //CKZIN DD *
|     DB2UPDATE -

```

```

DB2-HLQS(PROD1, TEST1, PROD2, TEST2 ) -
DDF( ... ) -
HLQ-NOT-UPDATED(RC(22)) -
SYSVALUE-DDN(SYSVALUE) -
JOURNAL-DDN(JOURNAL)
/*

```

Step 10

This step creates the conditional restart record. This example is for a non-data sharing subsystem. The required point in time is 2013/07/17 12:00:00.

Partial JCL for this step follows:

```

//S1 EXEC PGM=DSNJU003,REGION=6M
//STEPLIB DD DISP=SHR,DSN=DSN.VB10.SDSNLOAD
//SYSUT1 DD DISP=SHR,DSN=TEST1.BSDS01
//SYSUT2 DD DISP=SHR,DSN=TEST1.BSDS02
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
CRESTART CREATE,SYSPITRT=2013198120000
/*

```

Step 11

This step starts the DB2 system in special mode. This example is for a non-data sharing subsystem. Sample JCL can be found in the installation library SCKZJCL in member CKZDSTA.

Partial JCL for this step follows:

```

//S1 EXEC PGM=CKZ00010,REGION=6M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
// DD DISP=SHR,DSN=db2.SDSNEXIT
// DD DISP=SHR,DSN=db2.SDSNLOAD
//CKZINI DD DISP=SHR,DSN=HLQ?.SCKZPARM(CKZINI)
//CKZPRINT DD SYSOUT=*
//CKZIN DD *
DB2START -
SSID(DB2T) -
REPLY-TO-RESTART-WTOR(Y) -
DSNZPARM(specparm) -
SPECIAL
/*

```

Step 12

This step runs the RESTORE SYSTEM LOGONLY SWITCH VCAT. The SYSTEM RESTORE utility will apply log records to bring the target DB2 system to the desired point in time.

Partial JCL for this step follows:

```

//STEP1 EXEC DSNUPROC,TIME=1440,
// UTPROC='',
// SYSTEM='DB2T'
//SYSIN DD *
RESTORE SYSTEM LOGONLY SWITCH VCAT SYSVALUEDDN(SYSVALUE)
/*
//SYSVALUE DD DISP=SHR,DSN=HLQ?.WRK.SYSVALUE
/*

```

Step 13

This step stops the DB2 system. This example is for a non-data sharing subsystem. Sample JCL can be found in the installation library SCKZJCL in member CKZDSTO.

Partial JCL for this step follows:

```
//S1 EXEC PGM=CKZ00010,REGION=6M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
// DD DISP=SHR,DSN=db2.SDSNEXIT
// DD DISP=SHR,DSN=db2.SDSNLOAD
//CKZINI DD DISP=SHR,DSN=HLQ?.SCKZPARM(CKZINI)
//CKZPRINT DD SYSOUT=*
//CKZIN DD *
        DB2STOP -
            SSID(DB2T)
//*
```

Step 14

This step runs DB2UPDATE DBD01ONLY to reapply updates to the DB2 directory that may have been regressed by the RESTORE SYSTEM. Sample JCL can be found in the installation library SCKZJCL in member CKZDUPD.

Partial JCL for this step follows:

```
//S1 EXEC PGM=CKZ00010,REGION=6M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI DD DISP=SHR,DSN=HLQ?.SCKZPARM(CKZINI)
//CKZPRINT DD SYSOUT=*
//JOURNAL DD DISP=SHR,DSN=HLQ?.JRNL
//SYSDBDXA DD DISP=OLD,
// DSN=TEST1.DSNDBC.DSNDB01.SYSDBDXA.I0001.A001
//CKZIN DD *
        DB2UPDATE -
            DBD01ONLY -
            DB2-HLQS(PROD1, TEST1, PROD2, TEST2 ) -
            DDF( ... ) -
            HLQ-NOT-UPDATED(RC(22)) -
            JOURNAL-DDN(JOURNAL)
//*
```

Step 15

This step starts the DB2 system in special mode. Sample JCL can be found in the installation library SCKZJCL in member CKZDSTA.

Partial JCL for this step follows:

```
//S1 EXEC PGM=CKZ00010,REGION=6M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
// DD DISP=SHR,DSN=db2.SDSNEXIT
// DD DISP=SHR,DSN=db2.SDSNLOAD
//CKZINI DD DISP=SHR,DSN=HLQ?.SCKZPARM(CKZINI)
//CKZPRINT DD SYSOUT=*
//CKZIN DD *
        DB2START -
            SSID(DB2T) -
            DSNZPARM(specparm) -
            SPECIAL
//*
```


Step 16

This step runs DB2FIX DATABASES(DB2) to resolve any LPL or GRECP status on the DB2 directory and catalog. Sample JCL can be found in the installation library SCKZJCL in member CKZDFIX.

Partial JCL for this step follows:

```
//S1      EXEC PGM=CKZ00010,REGION=6M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//        DD DISP=SHR,DSN=db2.SDSNEXIT
//        DD DISP=SHR,DSN=db2.SDSNLOAD
//CKZINI  DD DISP=SHR,DSN=HLQ?.SCKZPARM(CKZINI)
//CKZPRINT DD SYSOUT=*
//CKZIN   DD *
          DB2FIX -
          SSID(DB2T) -
          DATABASES(DB2)
//*
```

If DB2FIX indicates that page space SYSDBDXA in database DSNDB01 was started due to having LPL or GRECP status, as indicated by the return code and by the message CKZ23526W DSNDB01.SYSDBDXA IS IN RESTRICTED STATUS; DB2UPDATE NEEDS TO BE RUN AGAIN, the changes made to SYSDBDXA by DB2UPDATE may have been regressed and need to be redone. The target DB2 subsystem must be stopped, DB2UPDATE run again using the DBD01ONLY keyword, and the DB2 subsystem started again in maintenance mode using the special zparms.

Step 17

This step runs DB2SQL to update the DB2 catalog. Sample JCL can be found in the installation library SCKZJCL in member CKZDSQL.

Partial JCL for this step follows:

```
//S1      EXEC PGM=CKZ00010,REGION=6M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI  DD DISP=SHR,DSN=HLQ?.SCKZPARM(CKZINI)
//CKZPRINT DD SYSOUT=*
//JOURNAL DD DISP=SHR,DSN=HLQ?.JRNL
//CKZIN   DD *
          DB2SQL -
          SSID(DB2T) -
          WLM-ENVIRONMENT-MASKS(PROD* TEST*) -
          JOURNAL-DDN(JOURNAL)
//*
```

Step 18

This step runs DB2FIX DATABASES(APPLICATION) to resolve any LPL or GRECP status on the application table and index spaces. Sample JCL can be found in the installation library SCKZJCL in member CKZDFIX.

Partial JCL for this step follows:

```
//S1      EXEC PGM=CKZ00010,REGION=6M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//        DD DISP=SHR,DSN=db2.SDSNEXIT
//        DD DISP=SHR,DSN=db2.SDSNLOAD
//CKZINI  DD DISP=SHR,DSN=HLQ?.SCKZPARM(CKZINI)
//CKZPRINT DD SYSOUT=*
//CKZIN   DD *
```

```

DB2FIX -
  SSID(DB2T) -
  DATABASES(APPLICATION)
//*

```

Step 19

This step stops the DB2 system. Sample JCL can be found in the installation library SCKZJCL in member CKZDSTO.

Partial JCL for this step follows:

```

//S1 EXEC PGM=CKZ00010,REGION=6M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
// DD DISP=SHR,DSN=db2.SDSNEXIT
// DD DISP=SHR,DSN=db2.SDSNLOAD
//CKZINI DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//CKZIN DD *
DB2STOP -
  SSID(DB2T)
//*

```

Step 20

This step is optional. This step is only used for the cloning of a DB2 system when it is desired to remove utility information from the target DB2 subsystem. This step will remove all entries from SYSUTILX. Sample JCL can be found in the installation library SCKZJCL in member CKZDUTCL.

Partial JCL for this step follows:

```

//S1 EXEC PGM=CKZ00010,REGION=6M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
// DD DISP=SHR,DSN=db2.SDSNEXIT
// DD DISP=SHR,DSN=db2.SDSNLOAD
//CKZINI DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//JOURNAL DD DISP=SHR,DSN=HLQ?.JRNL
//CKZIN DD *
DB2UTILXCLEAN -
  JOURNAL-DDN(JOURNAL)
//*

```

Step 21

This step starts the DB2 system in normal mode. This example is for a non-data sharing subsystem. Sample JCL can be found in the installation library SCKZJCL in member CKZDSTA.

Partial JCL for this step follows:

```

//S1 EXEC PGM=CKZ00010,REGION=6M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
// DD DISP=SHR,DSN=db2.SDSNEXIT
// DD DISP=SHR,DSN=db2.SDSNLOAD
//CKZINI DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//CKZIN DD *
DB2START -
  SSID(DB2T) -
  NORMAL
//*

```

Considerations for data sharing

To perform these cloning steps to a specific point in time from a DB2 BACKUP SYSTEM backup in a data sharing environment, note the following:

- Step 9 runs DB2UPDATE for all members. The DB2UPDATE for the secondary members should include BSDSONLY.
- Step 10 creates conditional restart records in each member.
- Step 11 starts all members in MAINT mode. The DB2START commands should include the keyword STOP-WAITING-IF-DSNR020I(Y,RC(3)).
- Step 13 stops all members.
- In step 15, only one member must be started.
- Step 21 starts all members in normal mode.

DB2 subsystem cloning from DB2 BACKUP SYSTEM dump tapes

DB2 subsystems can be cloned by using the backup dump tapes created by a DB2 BACKUP SYSTEM command.

Overview

Because the source DB2 is running at the time of the DB2 BACKUP SYSTEM command, this is an online cloning. This situation is similar to cloning using an interim set of volumes. DB2 Cloning Tool supports this process, but requires knowledge of the original source and final target volumes.

If the target volumes are set up in storage groups in a way that mirrors the source volume configuration, it is possible to pair the backup volumes to the target volumes by using the source and target storage group names. For example, if the source has two storage groups, one for logs (SRCSGL) and one for data (SRCSGD), and the target has two storage groups, one for logs (TGTSGL) and one for data (TGTSGD), then the dump tapes can be paired to the target volumes by using the source and target storage group names. The dump tapes that correspond to the source log storage group (SRCSGL) are paired with volumes in the target log storage group (TGTSGL), and the dump tapes that correspond to the source data storage group (SRCSGD) are paired with volumes in the target data storage group (TGTSGD).

The following procedure allows for repetitive cloning without requiring a manual update of the volume specifications when the LAST keyword is used with the DB2GETBACKINFO command.

- The source volumes (referred to as set A) have been copied to the backup dump tapes (set T) by a DB2 BACKUP SYSTEM command.
- The dump tapes (set T) are restored to the target volumes (set C), re-labeled with the target volume volser, and brought back online.
- The data sets on the target volumes (set C) are renamed.
- The target DB2 subsystems on the target volumes are conditioned.
- DB2 Cloning Tool requires the original source volumes (set A) and the final target volumes (set C) to rename and catalog the target volume data sets and to update the target DB2 system to replace the original source volumes with the final target volumes. To accomplish this, DB2 Cloning Tool must know the pairing between the source volumes (set A) and the target volumes (set C).

- To obtain the original source volumes (set A) and the backup dump tapes (set T) from DFSMSHsm, the DB2GETBACKINFO command is executed to query DFSMSHsm. Any generation can be selected in the DB2GETBACKINFO command. In this example, LAST is used for the most current version pairs of set A and set T. This list is provided in DFSMSHsm LIST format in HSMLIST DD. DB2 Cloning Tool also creates the BACKINFO DD data set in a format that is consistent with a utility in DB2 Recovery Expert and Rocket Mainstar DBR for DB2 on z/OS, only for the purpose of keeping cloning from a system backup consistent across products. The HSMLIST DD is not used again. The source ICF catalogs must also be identified by the user in this step.
- The job that executes the RESTORE-FROM-DUMPTAPES command does several things:
 - Reads the BACKINFO DD data set created in the previous step to:
 - Pair target volumes using TO-VOLSER or TO-STORAGEGROUP, along with volume capacity and the order in which the volumes (or storage groups) are specified.
 - Build and run ADRDSSU RESTORE commands to restore all eligible Source volumes.
 - Selects target volumes using TO-VOLSER and/or TO-STORAGEGROUP, along with volume capacity and the order in which the volumes (or storage groups) are specified to match volumes.
 - Creates the VOLPAIRS DD data set with volume pairs as determined above. These files will be used in the subsequent COPY step.
 - Creates the STATUS file to provide resume capability.
 - The user catalogs are added to this step and stored in the data set pointed to by UCATS DDN so that they can be passed to the COPY command.

For this procedure, one COPY command is used. The COPY command:

- Uses the DATA-MOVER(PGM(NONE)) keyword, which does not copy any volumes; the target volumes are created by the RESTORE-FROM-DUMPTAPES command, which restores the backup dump tapes onto the target volumes.
- Identifies the source volume (set A) to target volume (set C) pairing by reading the VOLPAIRS-DDN created in the previous step.
- Backs up the source ICF catalog copies from the target volumes.
- Creates a journal data set, which is used throughout the RENAME and DB2 conditioning steps.

The RENAME renames and catalogs all of the target volume data sets (set C) to new names.

The DB2 conditioning updates the appropriate DB2 data sets on the target volumes (set C) to the new data set names and volume serials.

If either an active log is defined with more than one stripe or it is a data sharing group, the active logs must be truncated at the point when the DB2 BACKUP SYSTEM FlashCopy of the database volumes completed prior to the first start. This can be accomplished by running the DB2ALTERBSDS command with the SLB-START keyword after DB2UPDATE has run and before DB2START is run. For data sharing, the DB2ALTERBSDS command with the SLB-START keyword should be run for each member of the target data sharing group.

For data sharing, the first members started will receive a DSNR020I WTOR due to the conditional restart records contained in the other members. If the start jobs are

being run serially by a job scheduler, it is recommended that the DB2START command for all but the last member include the STOP-WAITING-IF-DSNR020I keyword with a value of Y. This will allow the starts of the other members to be done concurrently. After the last member has been started, a DB2START for the first member should be done again using the WAITONLY and DB2-ALREADY-RUNNING(RC(0)) keywords. Completion of the DB2START with a return code of 4 or less indicates that the first member is up and ready for the remainder of the DB2 conditioning jobs.

Step overview

This section summarizes the steps you need to follow to perform DB2 subsystem cloning using the DB2 BACKUP SYSTEM dump tapes.

1. Run DB2GETBACKINFO to get the source (set A) and backup dump tape (set T) volume pairing used by the DB2 BACKUP SYSTEM command. (Any available generation can be specified; our example uses LAST.)
2. Run RESTORE-FROM-DUMPTAPES to take the backinfo data set created by the DB2GETBACKINFO command (Step 1) and generate and run ADRDSSU restore commands, and create metadata for use by the subsequent COPY command (Step 3).
3. Run COPY with DATA-MOVER(PGM(NONE)). Use VOLPAIRS-DDN to get the source volumes (set A) to target volumes (set C) pairing (data set from step 2), and use a USERCATALOGS-DDN keyword (data set from step 2).
4. Run RENAME for target volumes (set C).
5. Run the DB2 conditioning commands: DB2UPDATE, DB2START, DB2FIX DATABASES(DB2), DB2SQL, DB2FIX DATABASES(APPLICATION), DB2STOP.

Example

In the example steps that follow, the following items are used:

- The source DB2 resides on the source volumes (SRCxxx). The source log volumes are in storage group SRCSGL and the source data volumes are in storage group SRCSGD.
- The target DB2 will reside on the target volumes (TGTxxx). The target log volumes are in storage group TGTSGL and the target data volumes are in storage group TGTSGD.
- There are one or more sets of backup dump tapes (DMPxxx) that have been created by DB2 BACKUP SYSTEM commands.
- There are two source ICF catalogs (USERCAT.SRC01 and USERCAT.SRC02) that reside on source volumes.

Step 1

The first step is to get the source volume to backup dump tape pairing (SRCxxx to DMPxxx) and identify the source ICF catalogs from the last DB2 BACKUP SYSTEM taken for location DB2PLOC. Sample JCL can be found in the installation SCKZJCL library in member CKZDGETB.

Partial JCL for this step follows:

```
//S1      EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DSN=HLQ?.SCKZLOAD,DISP=SHR
//CKZINI  DD DSN=HLQ?.SCKZPARM(CKZINI),DISP=SHR
//CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
```

```

//BACKINFO DD DSN=HLQ?.WRK.BACKINFO,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          SPACE=(CYL,(1,1))
//HSMLIST  DD DSN=HLQ?.WRK.HSMLIST,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          SPACE=(CYL,(1,1))
//CKZIN    DD *
DB2GETBACKINFO          -
BACKINFO-DDN(BACKINFO) -
WORK-DDN(HSMLIST)      -
LAST                   -
LOCATION(DB2PLOC)        -
USE-DUMPTAPES          -
USERCATALOGS(          -
    USERCAT.SRC01      -
    USERCAT.SRC02      -
)
//*

```

Step 2

The second step is to restore the dump tapes of the source volumes to the target volumes or storage groups, using the output of Step 1 (BACKINFO data set). The user catalog pairs are also specified in this step. Sample JCL can be found in the installation SCKZJCL library in member CKZRSTDT.

Partial JCL for this step follows:

```

//S1      EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DSN=HLQ?.SCKZLOAD,DISP=SHR
//CKZINI  DD DSN=HLQ?.SCKZPARAM(CKZINI),DISP=SHR
//CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//BACKINFO DD DISP=SHR,DSN=HLQ?.WRK.BACKINFO
//VOLPAIRS DD DSN=HLQ?.WRK.VOLPAIRS,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          SPACE=(CYL,(1,1))
//STATUS  DD DSN=HLQ?.WRK.STATUS,
//          DISP=(,CATLG),SPACE=(CYL,(1,1)),
//          RECOG=KS,KEYLEN=64,KEYOFF=0,
//          LRECL=600,UNIT=SYSALLDA
//UCATS   DD DSN=HLQ?.WRK.UCATS,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          SPACE=(CYL,(1,1))
//CKZIN   DD *
RESTORE-FROM-DUMPTAPES          -
SOURCE-STORAGEGROUP( SRCGL SRCGD ) -
TO-STORAGEGROUP( TGTSGL TGTSGD ) -
BACKINFO-DDN(BACKINFO)        -
STATUS-DDN(STATUS)            -
VOLPAIRS-DDN(VOLPAIRS)        -
USERCATALOGS-DDN(UCATS)        -
USERCATALOGS(                  -
    USERCAT.SRC01 USERCAT.TGT01 -
    USERCAT.SRC02 USERCAT.TGT02 -
)
//*

```

Step 3

The third step is to set the pairing between source (SRCxxx) and target (TGTxxx) volumes in the journal and back up the source ICF catalogs from the target volumes. Sample JCL can be found in the installation SCKZJCL library in member CKZCOPY.

Partial JCL for this step follows:

```
//S1      EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI  DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//VOLPAIRS DD DISP=SHR,DSN=HLQ?.WRK.VOLPAIRS
//UCATS   DD DISP=SHR,DSN=HLQ?.WRK.UCATS
//JOURNAL DD DSN=HLQ?.JRNL,
//        DISP=(,CATLG),UNIT=SYSALLDA,
//        RECORG=KS,KEYLEN=64,KEYOFF=0,
//        LRECL=600,SPACE=(CYL,(10,10))
//CKZIN   DD *
COPY
  DATA-MOVER(PGM(NONE)) -
  VOLPAIRS-DDN(VOLPAIRS) -
  USERCATALOGS-DDN(UCATS) -
  CATWORK-DSN(HLQ?.WRK.* ) -
  JOURNAL-DDN(JOURNAL)
//*
```

Step 4

The fourth step is the RENAME of the data sets on the target volumes (TGTxxx). Sample JCL can be found in the installation SCKZJCL library in member CKZREN.

Partial JCL for this step follows:

```
//S1      EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI  DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//SORTMSG DD SYSOUT=*
//CKZPRINT DD SYSOUT=*
//DRSTATS DD SYSOUT=*
//JOURNAL DD DSN=HLQ?.JRNL,DISP=SHR
//BCSRECS DD DSN=HLQ?.WRK.BCSRECS,
//        UNIT=SYSALLDA,DISP=(,CATLG),
//        SPACE=(CYL,(10,10))
//VOLBKUP DD DSN=HLQ?.WRK.VOLBKUP,
//        UNIT=SYSALLDA,DISP=(,CATLG),
//        SPACE=(CYL,(40,40))
//CKZIN   DD *
RENAME
  SAFE -
  VOLBKUP-DDN(VOLBKUP) -
  RENAME-MASKS(
    PRODLG.** TESTLG.** -
    PROddb.** TESTDB.** -
  )
  JOURNAL-DDN(JOURNAL)
//*
```

Step 5

The last step is to run the DB2 conditioning commands. These commands are:

- DB2UPDATE
- DB2ALTERBSDS SLB-START
- DB2START
- DB2FIX DATABASES(DB2)
- DB2SQL
- DB2FIX DATABASES(APPLICATION)
- DB2STOP

For additional information about the conditioning commands, refer to the appropriate section in “DB2 online cloning” on page 104 for your specific type of cloning.

If either an active log is defined with more than one stripe or it is a data sharing group, the active logs must be truncated at the point when the DB2 BACKUP SYSTEM FlashCopy of the database volumes completed prior to the first start. This can be accomplished by running the DB2ALTERBSDS command with the SLB-START keyword after DB2UPDATE has run and before DB2START is run. For data sharing, the DB2ALTERBSDS command with the SLB-START keyword should be run for each member of the target data sharing group.

The following are examples of the DB2UPDATE, DB2ALTERBSDS, and DB2START steps necessary to handle this condition.

Example 1 – a non data sharing subsystem with active logs that are striped:

1. Update the DB2 directory and the BSDS.

```
DB2UPDATE                -
  DB2-HLQS( srcvcat tgtvcat ) -
  DDF( ... )              -
  JOURNAL-DDN(JOURNAL)
```

2. Create a system-level backup (SLB) start conditional restart record in the BSDS.

```
DB2ALTERBSDS            -
  SLB-START              -
  JOURNAL-DDN(JOURNAL)
```

3. Start the target DB2 in special mode and automatically reply to the DB2 restart WTOR.

```
DB2START                 -
  DB2-SSID(ssid)         -
  SPECIAL                 -
  DSNZPARM(ssidSPEC)     -
  REPLY-TO-RESTART-WTOR(Y)
```

Example 2 – a data sharing group with three members (the active logs may or may not be striped):

1. Update the DB2 directory and the BSDS for mbr1.

```
DB2UPDATE                -
  DB2-HLQS( srcvcat tgtvcat ) -
  DB2-GROUP( srcgrp tgtgrp ) -
  DB2-MEMBERS( srcmbr1 mbr1 -
               srcmbr2 mbr2 -
               srcmbr3 mbr3 ) -
  DDF( ... )              -
  JOURNAL-DDN(JOURNAL)
```

2. Update the BSDS for mbr2.

```
DB2UPDATE                -
  BSDSONLY                -
  DB2-HLQS( srcvcat tgtvcat ) -
  DB2-GROUP( srcgrp tgtgrp ) -
  DB2-MEMBERS( srcmbr1 mbr1 -
               srcmbr2 mbr2 -
               srcmbr3 mbr3 ) -
  DDF( ... )              -
  JOURNAL-DDN(JOURNAL)
```

3. Update the BSDS for mbr3.

```
DB2UPDATE                -
  BSDSONLY                -
  DB2-HLQS( srcvcat tgtvcat ) -
```



```

|         DB2-GROUP( srcgrp tgtgrp ) -
|         DB2-MEMBERS( srcmbr1 mbr1 -
|             srcmbr2 mbr2 -
|             srcmbr3 mbr3 ) -
|         DDF( ... ) -
|         JOURNAL-DDN(JOURNAL)

```

4. Create a system-level backup (SLB) start conditional restart record in the BSDS for mbr1.

```

|         DB2ALTERBSDS -
|         DB2-MEMBER(mbr1) -
|         SLB-START -
|         JOURNAL-DDN(JOURNAL)

```

5. Create a system-level backup (SLB) start conditional restart record in the BSDS for mbr2.

```

|         DB2ALTERBSDS -
|         DB2-MEMBER(mbr2) -
|         SLB-START -
|         JOURNAL-DDN(JOURNAL)

```

6. Create a system-level backup (SLB) start conditional restart record in the BSDS for mbr3.

```

|         DB2ALTERBSDS -
|         DB2-MEMBER(mbr3) -
|         SLB-START -
|         JOURNAL-DDN(JOURNAL)

```

7. Start the target DB2 mbr1 in special mode, automatically reply to the DB2 restart WTOR, and stop waiting if a DSNR020I WTOR is received.

```

|         DB2START -
|         DB2-SSID(mbr1) -
|         SPECIAL -
|         DSNZPARM(mbr1SPEC) -
|         REPLY-TO-RESTART-WTOR(Y) -
|         STOP-WAITING-IF-DSNR020I(Y)

```

8. Start the target DB2 mbr2 in special and light mode, automatically reply to the DB2 restart WTOR, and stop waiting if a DSNR020I WTOR is received.

```

|         DB2START -
|         DB2-SSID(mbr2) -
|         SPECIAL -
|         DSNZPARM(mbr2SPEC) -
|         LIGHT -
|         REPLY-TO-RESTART-WTOR(Y) -
|         STOP-WAITING-IF-DSNR020I(Y)

```

9. Start the target DB2 mbr3 in special and light mode and automatically reply to the DB2 restart WTOR.

```

|         DB2START -
|         DB2-SSID(mbr3) -
|         SPECIAL -
|         DSNZPARM(mbr3SPEC) -
|         LIGHT -
|         REPLY-TO-RESTART-WTOR(Y)

```

10. Wait for the target DB2 mbr1 to complete its startup.

```

|         DB2START -
|         DB2-SSID(mbr1) -
|         SPECIAL -
|         DSNZPARM(mbr1SPEC) -
|         REPLY-TO-RESTART-WTOR(Y) -
|         DB2-ALREADY-RUNNING(RC(0)) -
|         WAITONLY

```

DB2 subsystem cloning from other system level backups when backup volumes are online

DB2 subsystems can be cloned using the backup volumes created by one of the following backup products: DB2 Recovery Expert for z/OS, Rocket System Backup and Recovery for DB2 (RBR), or Database Backup & Recovery for DB2 (DBR for DB2). Because the source DB2 is running at the time of the backup, this is an online cloning.

For the purposes of this procedure, the following acronyms are used and should be substituted where the variable *ccc* is used:

- DB2 Recovery Expert: *ccc*=ARY
- RBR: *ccc*=RBR
- DBR for DB2: *ccc*=DBR

The following procedure allows for repetitive cloning without the need to manually update the volume specifications when LAST is used with the *ccc*#VOLS program. The *ccc*#VOLS program is part of the DB2 Recovery Expert, RBR, and DBR products.

- The source volumes (referred to as set A) have been copied to the backup volumes (set B) by a DB2 Recovery Expert, RBR, or DBR for DB2 backup.
- The backup volumes (set B) are copied to the target volumes (set C).
- The data sets on the target volumes (set C) are renamed.
- The target DB2 subsystems on the target volumes are conditioned.

For this procedure, the source to backup volume pairing will be obtained from DB2 Recovery Expert, RBR, or DBR for DB2 and reformatted for use by two COPY commands. Each COPY command must be in its own JCL step.

- The first COPY command identifies the source volume to backup volume pairing and backs up the source ICF catalog copies on the backup volumes. This COPY does not do any volume copies as the copies were done by the DB2 Recovery Expert, RBR, or DBR for DB2 backup.
- The second COPY command copies the backup volumes to the target volumes and does not back up the ICF catalogs.

An additional DD must be added to the COPY steps. The DD name is VOLPLIST and it must have attributes of RECFM=FB,LRECL=80. The data set allocated by the VOLPLIST DD will be used by a subsequent step/job.

Step overview

1. Run the *ccc*#VOLS program to get the source (set A) and backup (set B) volume pairing used by the DB2 Recovery Expert, RBR, or DBR for DB2 backup and the names of the source ICF user catalogs.
2. Run BACKINFO-REFORMAT to take the backinfo data set created by the *ccc*#VOLS program (Step 1) and reformat it for use by subsequent COPY commands (Steps 3 and 4).
3. Run COPY with DATA-MOVER(PGM(NONE)). Use VOLPAIRS-DDN to get the source volumes (set A) to backup volumes (set B) pairing (data set from step 2), use a USERCATALOGS-DDN keyword (data set from step 2), and include the VOLPLIST DD.
4. Run COPY using FROM-VOLSER-DDN to get the backup volumes (set B) (data set from step 2), using the NOUSERCATALOGS keyword and including the VOLPLIST DD.

5. Run CKZRNTGT with the VOLPLIST from step 3 on the CKZIN DD and the VOLPLIST from step 4 on the NUCIN DD.
6. Run VOOPTIONS using the NEWTGT data set from step 5 as input to NEWTARGETS-DDN.
7. Run RENAME for target volumes (set C).
8. Run the DB2 conditioning commands: DB2UPDATE, DB2START, DB2FIX DATABASES(DB2), DB2SQL, DB2FIX DATABASES(APPLICATION), DB2STOP, DB2UTILXCLEAN.

Note: Steps 3, 6, 7, and 8 use the same journal data set. Step 4 uses a different journal data set.

Example

In this example:

- The source DB2 resides on the source volumes (SRCxxx)
- The target DB2 will reside on the target volumes (TGTxxx).
- There are one or more sets of backup volumes (BKPxxx) that have been created by DB2 Recovery Expert, RBR, or DBR for DB2 backups.
- There are two source ICF catalogs (USERCAT.SRC01 and USERCAT.SRC02) that reside on source volumes.

Step 1 - Get the source to backup volume pairing (SRCxxx to BKPxxx) and the source ICF catalog names from the last backup taken for DB2 subsystem DB2P.

Run the *ccc#VOLS* program to get the source (set A) and backup (set B) volume pairing used by the DB2 Recovery Expert, RBR, or DBR for DB2 backup and the names of the source ICF user catalogs. Sample JCL for this program can be found in the DB2 Recovery Expert, RBR, or DBR for DB2 *ScccSAMP* library.

Step 2 - Reformat the output of step 1 (backinfo data set) for use in the COPY in steps 3 and 4.

The user catalog pairs are also specified here. Sample JCL can be found in the installation SCKZJCL library in member CKZBKIRF.

Partial JCL:

```
//S1      EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DSN=HLQ?.SCKZLOAD,DISP=SHR
//CKZINI  DD DSN=HLQ?.SCKZPARM(CKZINI),DISP=SHR
//CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//BACKINFO DD DISP=SHR,DSN=HLQ?.WRK.BACKINFO
//VOLPAIRS DD DSN=HLQ?.WRK.VOLPAIRS,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          SPACE=(CYL,(1,1))
//FRVOLSER DD DSN=HLQ?.WRK.FRVOLSER,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          SPACE=(CYL,(1,1))
//UCATS   DD DSN=HLQ?.WRK.UCATS,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          SPACE=(CYL,(1,1))
//CKZIN   DD *
          BACKINFO-REFORMAT           -
          BACKINFO-DDN(BACKINFO)      -
          VOLPAIRS-DDN(VOLPAIRS)      -
          FROM-VOLSER-DDN(FRVOLSER)   -
          USERCATALOGS-DDN(UCATS)     -
```

```

        USERCATALOGS(
            USERCAT.SRC01 USERCAT.TGT01
            USERCAT.SRC02 USERCAT.TGT02
        )
    /**

```

Step 3 - Set the pairing between source (SRCxxx) and backup (BKPxxx) volumes in the journal and back up the source ICF catalogs from the backup volumes.

Sample JCL can be found in the installation SCKZJCL library in member CKZCOPY.

Partial JCL:

```

//S1      EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI  DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//VOLPAIRS DD DISP=SHR,DSN=HLQ?.WRK.VOLPAIRS
//UCATS   DD DISP=SHR,DSN=HLQ?.WRK.UCATS
//JOURNAL DD DSN=HLQ?.JRNL,
//         DISP=(,CATLG),UNIT=SYSALLDA,
//         RECOG=KS,KEYLEN=64,KEYOFF=0,
//         LRECL=600,SPACE=(CYL,(10,10))
//VOLPLIST DD DSN=HLQ?.WRK.VOLPLIST,
//         DISP=(,CATLG),UNIT=SYSALLDA,
//         RECFM=FB,LRECL=80,BLKSIZE=0,
//         SPACE=(CYL,(1,1))
//CKZIN   DD *
COPY
    DATA-MOVER(PGM(NONE))
    VOLPAIRS-DDN(VOLPAIRS)
    USERCATALOGS-DDN(UCATS)
    CATWORK-DSN(HLQ?.WRK.* )
    JOURNAL-DDN(JOURNAL)
/**

```

Step 4 - Copy the backup volumes (BKPxxx) to the target volumes (TGTxxx) without backing up any ICF catalogs.

Sample JCL can be found in the installation SCKZJCL library in member CKZCOPY.

Partial JCL:

```

//S1      EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI  DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//FRVOLSER DD DISP=SHR,DSN=HLQ?.WRK.FRVOLSER
//JOURNAL DD DSN=HLQ?.NUCJRNL,
//         DISP=(,CATLG),UNIT=SYSALLDA,
//         RECOG=KS,KEYLEN=64,KEYOFF=0,
//         LRECL=600,SPACE=(CYL,(10,10))
//VOLPLIST DD DSN=HLQ?.NUC.WRK.VOLPLIST,
//         DISP=(,CATLG),UNIT=SYSALLDA,
//         RECFM=FB,LRECL=80,BLKSIZE=0,
//         SPACE=(CYL,(1,1))
//CKZIN   DD *
COPY
    FROM-VOLSER-DDN(FRVOLSER)
    TO-VOLSER(
        TGT001 TGT002
    )
    NOUSERCATALOGS
    JOURNAL-DDN(JOURNAL)
/**

```

Step 5 - Run CKZRNTGT with the VOLPLIST data sets from steps 3 and 4.

Sample JCL can be found in the installation sCKZJCL library in member CKZRNTGT.

Partial JCL:

```
//* CKZIN IS FROM THE DB2 Cloning Tool COPY COMMAND WITH A USERCATALOGS KEYWORD,
//*      DD VOLPLIST.
//* NUCIN IS FROM THE DB2 Cloning Tool COPY COMMAND WITH THE NOUSERCATALOGS KEYWORD,
//*      DD VOLPLIST.
//* NEWTGT WILL BE USED BY THE DB2 Cloning Tool VOLOPTIIONS COMMAND
//S2      EXEC PGM=IRXJCL,REGION=2M,PARM='CKZRNTGT'
//SYSEXEC DD DSN=HLQ?.SCKZPARM,DISP=SHR
//SYSTSIN DD DUMMY
//SYSPRINT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*,DCB=(LRECL=132,RECFM=VBA,BLKSIZE=0)
//CKZIN   DD DSN=HLQ?.WRK.VOLPLIST,DISP=SHR
//NUCIN   DD DSN=HLQ?.NUC.WRK.VOLPLIST,DISP=SHR
//NEWTGT  DD DSN=HLQ?.WRK.NEWTGT,UNIT=SYSDA,DISP=(,CATLG),
//          DSORG=PS,LRECL=80,RECFM=FB,BLKSIZE=0,
//          SPACE=(CYL,(1,1))
```

Step 6 - Run VOLOPTIIONS with the newtgt data set from step 5 to update the journal with the actual source (SRCxxx) to target (TGTxxx) volume pairing for RENAME.

Sample JCL can be found in the installation SCKZJCL library in member CKZVOLOP.

Partial JCL:

```
//S1      EXEC PGM=CKZ00010,REGION=6M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI  DD DISP=SHR,DSN=HLQ?.SCKZPARM(CKZINI)
//CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//JOURNAL DD DISP=SHR,DSN=HLQ?.JRNL
//NEWTGT  DD DSN=HLQ?.WRK.NEWTGT,UNIT=SYSDA,DISP=SHR
//CKZIN   DD *
          VOLOPTIIONS UPDATE          -
          NEWTARGETS-DDN(NEWTGT)      -
          JOURNAL-DDN(JOURNAL)
//*
```

Step 7 - RENAME the data sets on the target volumes (TGTxxx).

Sample JCL can be found in the installation SCKZJCL library in member CKZREN.

Partial JCL and command:

```
//S1      EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI  DD DISP=SHR,DSN=HLQ?.SCKZPARM(CKZINI)
//SORTMSG DD SYSOUT=*
//CKZPRINT DD SYSOUT=*
//DRSTATS DD SYSOUT=*
//JOURNAL DD DSN=HLQ?.JRNL,DISP=SHR
//BCSRECS DD DSN=HLQ?.WRK.BCSRECS,
//          UNIT=SYSALLDA,DISP=(,CATLG),
//          SPACE=(CYL,(10,10))
//VOLBKUP DD DSN=HLQ?.WRK.VOLBKUP,
//          UNIT=SYSALLDA,DISP=(,CATLG),
//          SPACE=(CYL,(40,40))
//CKZIN   DD *
          RENAME                      -
          SAFE                        -
          VOLBKUP-DDN(VOLBKUP)        -
```

```

        RENAME-MASKS (
                PROD1.** TEST1.** -
                PROD2.** TEST2.** -
        ) -
    JOURNAL-DDN(JOURNAL)
//*

```

Step 8 - Run the DB2 conditioning commands.

Run the DB2 conditioning commands: DB2UPDATE, DB2START, DB2FIX DATABASES(DB2), DB2SQL, DB2FIX DATABASES(APPLICATION), DB2STOP, DB2UTILXCLEAN. For additional information about the conditioning commands, refer to the “DB2 online cloning” on page 104 section for your specific type of cloning.

DB2 subsystem cloning from other system level backups when backup volumes are offline

DB2 subsystems can be cloned using the backup volumes created by one of the following backup products: DB2 Recovery Expert for z/OS, Rocket System Backup and Recovery for DB2 (RBR), or Database Backup & Recovery for DB2 (DBR for DB2). The offline backup volumes are exact copies of the source volumes and have an internal volser that is the same as a source volser. Because the source DB2 is running at the time of the backup, this is an online cloning.

For the purposes of this procedure, the following acronyms are used and should be substituted where the variable *ccc* is used:

- DB2 Recovery Expert: *ccc*=ARY
- RBR: *ccc*=RBR
- DBR for DB2: *ccc*=DBR

The following procedure allows for repetitive cloning without the need to manually update the volume specifications when LAST is used with the *ccc*#VOLS program. The *ccc*#VOLS program is part of the DB2 Recovery Expert, RBR, and DBR products.

- The source volumes (referred to as set A) have been copied to the backup volumes (set B) by a DB2 Recovery Expert, RBR, or DBR for DB2 backup.
- The backup volumes (set B) are clipped with unique volsers and varied online.
- The backup volumes (set B) are copied to the target volumes (set C).
- The backup volumes (set B) are varied offline and clipped back to their original source volser.
- The data sets on the target volumes (set C) are renamed.
- The target DB2 subsystems on the target volumes are conditioned.

For this procedure, the source to backup volume pairing will be obtained from DB2 Recovery Expert, RBR, or DBR for DB2 and reformatted for use by two COPY commands. Each COPY command must be in its own JCL step.

- The first COPY command identifies the source volume to backup volume pairing and backs up the source ICF catalog copies on the backup volumes. This COPY does not do any volume copies as the copies were done by the DB2 Recovery Expert, RBR, or DBR for DB2 backup.
- The second COPY command copies the backup volumes to the target volumes and does not back up the ICF catalogs.

An additional DD must be added to the COPY steps. The DD name is VOLPLIST and it must have attributes of RECFM=FB,LRECL=80. The data set allocated by the VOLPLIST DD will be used by a subsequent step/job.

Step overview

1. Run the *ccc#VOLS* program to get the source (set A) and backup (set B) volume pairing used by the DB2 Recovery Expert, RBR, or DBR for DB2 backup and the names of the source ICF user catalogs.
2. Run BACKINFO-REFORMAT to take the backinfo data set created by the *ccc#VOLS* program (Step 1) and reformat it for use by subsequent COPY commands (Steps 3 and 4). The CLIP-IF-OFFLINE(Y) is used to request that the offline backup volumes (set B) be clipped and varied online. The VOLSER-RENAME-MASKS keyword is used to specify how the source volsers should be changed to generate unique volsers for the corresponding backup volumes being clipped. The VOLOPTIONS-CMD-DDN keyword is specified to generate a VOLOPTIONS command to unclip the backup volumes after they have been copied to the target volumes.
3. Run COPY with DATA-MOVER(PGM(NONE)). Use VOLPAIRS-DDN to get the source volumes (set A) to backup volumes (set B) pairing (data set from step 2), use a USERCATALOGS-DDN keyword (data set from step 2), and include the VOLPLIST DD.
4. Run COPY using FROM-VOLSER-DDN to get the backup volumes (set B) (data set from step 2), using the NOUSERCATALOGS keyword and including the VOLPLIST DD.
5. Run VOLOPTIONS with CKZIN using the data set created in step 2 by the VOLOPTIONS-CMD-DDN keyword to vary offline and unclip the backup volumes (set B). The journal data set created in step 3 is used.
6. Run CKZRNTGT with the VOLPLIST from step 3 on the CKZIN DD and the VOLPLIST from step 4 on the NUCIN DD.
7. Run VOLOPTIONS using the NEWTGT data set from step 6 as input to NEWTARGETS-DDN.
8. Run RENAME for target volumes (set C).
9. Run the DB2 conditioning commands: DB2UPDATE, DB2START, DB2FIX DATABASES(DB2), DB2SQL, DB2FIX DATABASES(APPLICATION), DB2STOP, DB2UTILXCLEAN.

Note: Steps 3, 5, 7, 8, and 9 use the same journal data set. Step 4 uses a different journal data set.

Example

In this example:

- The source DB2 resides on the source volumes (SRCxxx)
- The target DB2 will reside on the target volumes (TGTxxx).
- There are one or more sets of backup volumes (BKPxxx) that have been created by DB2 Recovery Expert, RBR, or DBR for DB2 backups. The backup volumes are offline and have internal volsers of their corresponding source volumes.
- There are two source ICF catalogs (USERCAT.SRC01 and USERCAT.SRC02) that reside on source volumes.

Step 1 - Get the source to backup volume pairing (SRCxxx to BKPxxx) and the source ICF catalog names from the last backup taken for DB2 subsystem DB2P.

Run the *ccc#VOLS* program to get the source (set A) and backup (set B) volume pairing used by the DB2 Recovery Expert, RBR, or DBR for DB2 backup and the names of the source ICF user catalogs. Sample JCL for this program can be found in the DB2 Recovery Expert, RBR, or DBR for DB2 *ScccSAMP* library.

Step 2 - Reformat the output of step 1 (backinfo data set) for use in the COPY in steps 3 and 4.

The offline backup volumes are clipped to new volsers and varied online. The user catalog pairs are also specified here. Sample JCL can be found in the installation *SCKZJCL* library in member *CKZBKIRF*.

Partial JCL:

```
//S1      EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DSN=HLQ?.SCKZLOAD,DISP=SHR
//CKZINI  DD DSN=HLQ?.SCKZPARM(CKZINI),DISP=SHR
//CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//BACKINFO DD DISP=SHR,DSN=HLQ?.WRK.BACKINFO
//VOLPAIRS DD DSN=HLQ?.WRK.VOLPAIRS,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          SPACE=(CYL,(1,1))
//FRVOLSER DD DSN=HLQ?.WRK.FRVOLSER,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          SPACE=(CYL,(1,1))
//UCATS   DD DSN=HLQ?.WRK.UCATS,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          SPACE=(CYL,(1,1))
//VOLOPCMD DD DSN=HLQ?.WRK.VOLOPCMD,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          SPACE=(TRK,(1,1))
//CKZIN   DD *
BACKINFO-REFORMAT          -
BACKINFO-DDN(BACKINFO)    -
VOLPAIRS-DDN(VOLPAIRS)    -
FROM-VOLSER-DDN(FRVOLSER) -
CLIP-IF-OFFLINE(Y)        -
VOLSER-RENAME-MASKS(      -
  SRC* BKP*                -
)                            -
VOLOPTIONS-CMD-DDN( VOLOPCMD ) -
USERCATALOGS-DDN(UCATS)   -
USERCATALOGS(             -
  USERCAT.SRC01 USERCAT.TGT01 -
  USERCAT.SRC02 USERCAT.TGT02 -
)                            -
/**
```

Step 3 - Set the pairing between source (SRCxxx) and backup (BKPxxx) volumes in the journal and back up the source ICF catalogs from the backup volumes.

Sample JCL can be found in the installation *SCKZJCL* library in member *CKZCOPY*.

Partial JCL:

```
//S1      EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI  DD DISP=SHR,DSN=HLQ?.SCKZPARM(CKZINI)
//CKZPRINT DD SYSOUT=*
```



```

//VOLPAIRS DD DISP=SHR,DSN=HLQ?.WRK.VOLPAIRS
//UCATS DD DISP=SHR,DSN=HLQ?.WRK.UCATS
//JOURNAL DD DSN=HLQ?.JRNL,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          RECORG=KS,KEYLEN=64,KEYOFF=0,
//          LRECL=600,SPACE=(CYL,(10,10))
//VOLPLIST DD DSN=HLQ?.WRK.VOLPLIST,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          RECFM=FB,LRECL=80,BLKSIZE=0,
//          SPACE=(CYL,(1,1))
//CKZIN DD *
COPY
  DATA-MOVER(PGM(NONE)) -
  VOLPAIRS-DDN(VOLPAIRS) -
  USERCATALOGS-DDN(UCATS) -
  CATWORK-DSN(HLQ?.WRK.* ) -
  JOURNAL-DDN(JOURNAL)
//*

```

Step 4 - Copy the backup volumes (BKPxxx) to the target volumes (TGTxxx) without backing up any ICF catalogs.

Sample JCL can be found in the installation SCKZJCL library in member CKZCOPY.

Partial JCL:

```

//S1 EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//FRVOLSER DD DISP=SHR,DSN=HLQ?.WRK.FRVOLSER
//JOURNAL DD DSN=HLQ?.NUCJRNL,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          RECORG=KS,KEYLEN=64,KEYOFF=0,
//          LRECL=600,SPACE=(CYL,(10,10))
//VOLPLIST DD DSN=HLQ?.NUC.WRK.VOLPLIST,
//          DISP=(,CATLG),UNIT=SYSALLDA,
//          RECFM=FB,LRECL=80,BLKSIZE=0,
//          SPACE=(CYL,(1,1))
//CKZIN DD *
COPY
  FROM-VOLSER-DDN(FRVOLSER) -
  TO-VOLSER(
    TGT001 TGT002
  )
  NOUSERCATALOGS
  JOURNAL-DDN(JOURNAL)
//*

```

Step 5 - Run VOLOPTIONS command created in step 2 to vary offline and unclip the backup volumes.

Sample JCL can be found in the installation SCKZJCL library in member CKZVOLOP.

Partial JCL:

```

//S1 EXEC PGM=CKZ00010,REGION=6M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI DD DISP=SHR,DSN=HLQ?.SCKZPARAM(CKZINI)
//CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//JOURNAL DD DISP=SHR,DSN=HLQ?.JRNL
//CKZIN DD DISP=SHR,DSN=HLQ?.WRK.VOLOPCMD
//*

```

Step 6 - Run CKZRNTGT with the VOLPLIST data sets from steps 3 and 4.

Sample JCL can be found in the installation SCKZJCL library in member CKZRNTGT.

Partial JCL:

```
//* CKZIN IS FROM THE DB2 Cloning Tool COPY COMMAND WITH A USERCATALOGS KEYWORD,
//*      DD VOLPLIST.
//* NUCIN IS FROM THE DB2 Cloning Tool COPY COMMAND WITH THE NOUSERCATALOGS KEYWORD,
//*      DD VOLPLIST.
//* NEWTGT WILL BE USED BY THE DB2 Cloning Tool VOLOPTIIONS COMMAND
//S2     EXEC PGM=IRXJCL,REGION=2M,PARM='CKZRNTGT'
//SYSEXEC DD DSN=HLQ?.SCKZPARM,DISP=SHR
//SYSTSIN DD DUMMY
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*,DCB=(LRECL=132,RECFM=VBA,BLKSIZE=0)
//CKZIN  DD DSN=HLQ?.WRK.VOLPLIST,DISP=SHR
//NUCIN  DD DSN=HLQ?.NUC.WRK.VOLPLIST,DISP=SHR
//NEWTGT DD DSN=HLQ?.WRK.NEWTGT,UNIT=SYSDA,DISP=(,CATLG),
//        DSORG=PS,LRECL=80,RECFM=FB,BLKSIZE=0,
//        SPACE=(CYL,(1,1))
```

Step 7 - Run VOLOPTIIONS with the newtgt data set from step 5 to update the journal with the actual source (SRCxxx) to target (TGTxxx) volume pairing for RENAME.

Sample JCL can be found in the installation SCKZJCL library in member CKZVOLOP.

Partial JCL:

```
//S1     EXEC PGM=CKZ00010,REGION=6M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI  DD DISP=SHR,DSN=HLQ?.SCKZPARM(CKZINI)
//CKZPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//JOURNAL DD DISP=SHR,DSN=HLQ?.JRNL
//NEWTGT  DD DSN=HLQ?.WRK.NEWTGT,UNIT=SYSDA,DISP=SHR
//CKZIN   DD *
          VOLOPTIIONS UPDATE           -
          NEWTARGETS-DDN(NEWTGT)      -
          JOURNAL-DDN(JOURNAL)
//*
```

Step 8 - RENAME the data sets on the target volumes (TGTxxx).

Sample JCL can be found in the installation SCKZJCL library in member CKZREN.

Partial JCL and command:

```
//S1     EXEC PGM=CKZ00010,REGION=8M
//STEPLIB DD DISP=SHR,DSN=HLQ?.SCKZLOAD
//CKZINI  DD DISP=SHR,DSN=HLQ?.SCKZPARM(CKZINI)
//SORTMSG DD SYSOUT=*
//CKZPRINT DD SYSOUT=*
//DRSTATS DD SYSOUT=*
//JOURNAL DD DSN=HLQ?.JRNL,DISP=SHR
//BCSRECS DD DSN=HLQ?.WRK.BCSRECS,
//        UNIT=SYSALLDA,DISP=(,CATLG),
//        SPACE=(CYL,(10,10))
//VOLBKUP DD DSN=HLQ?.WRK.VOLBKUP,
//        UNIT=SYSALLDA,DISP=(,CATLG),
//        SPACE=(CYL,(40,40))
//CKZIN   DD *
          RENAME                       -
          SAFE                          -
          VOLBKUP-DDN(VOLBKUP)         -
```

```

        RENAME-MASKS (
                PROD1.** TEST1.** -
                PROD2.** TEST2.** -
        ) -
    JOURNAL-DDN(JOURNAL)
//*
```

Step 9 - Run the DB2 conditioning commands.

Run the DB2 conditioning commands: DB2UPDATE, DB2START, DB2FIX DATABASES(DB2), DB2SQL, DB2FIX DATABASES(APPLICATION), DB2STOP, DB2UTILXCLEAN. For additional information about the conditioning commands, refer to the “DB2 online cloning” on page 104 section for your specific type of cloning.

CKZINI customization values

The CKZINI member (the product initialization member) defines global information regarding DB2 Cloning Tool usage and options within your installation. This topic provides guidelines, values, and syntax used in the CKZINI member.

Structure of the CKZINI member

The CKZINI member is organized in sections. Each section contains a set of individual parameter specifications known as tokens. A single CKZINI member is recommended for use by all users of DB2 Cloning Tool at the installation site.

Syntax rules

The CKZINI consists of token assignment statements that are organized by sections. The general format for a token statement is:

```
Keyword = Parameter(s)
```

Leading blanks from the beginning of the logical card-image records are allowed for all statements. Syntax scan processing locates the first non-blank character in each logical record. The entire length of the 80-byte logical record is considered for valid data. Do not renumber the CKZINI member; sequence numbers assigned in columns 73 through 80 will cause errors.

You may add a notes token to document your changes to any section including the initial/unnamed section. The INIMERGE program will retain these notes along with any comments that continue from the notes token.

```
Notes = Updated by Dan on 2011/04/23 +
        Updated by John on 2011/02/20
```

Section names

A *section name* is indicated by the colon character (:), immediately followed by the section name string. Section names are to be changed or added only under the direction of IBM Software Support.

For a multi-image CKZINI, the section name can be qualified by sysplex and/or system name, except for the PRODUCT_INFO and INIMERGE_VALUES sections:

Sysplex-name and system-name qualification:
:section-name.sysplex-name.system-name

Sysplex-name only qualification:

:section-name.sysplex-name

System-name only qualification:

:section-name..system-name

The sysplex and system names must be explicitly specified with no wild-card characters allowed.

INIMERGE will retain your qualified sections and merge new tokens for those sections.

DB2 Cloning Tool will use only the first matching section in the CKZINI member. All other variations of that section will be ignored. You may specify qualified sections for those images with unique requirements, followed by an unqualified section that applies to all of the other images.

Token name

A *token name* is a keyword value that can be specified under particular section names. They are specified as a blank-delimited character string to the left of an equal (=) sign. Token names are to be changed or added only under the direction of IBM Software Support except where noted.

Token value

A *token value* is data that is specified as a set of strings to the right of the equal (=) sign after a token name. Token values may be keywords, user values, or a keyword with a token value, shown as KEYWORD(uservalue).

Values may be enclosed in either single or double quotes. The quotes are stripped away before the product uses the value. Quotes may be supplied as data by:

- Using the opposite quote symbol as delimiters (e.g., " " will yield a single quote as data).
- Specifying two to get one (e.g., ' ' will yield a single quote as data).

Continuation rules

Statements can be continued, using either a minus (-) or plus (+) character anywhere within the text. All data to the right of the continuation character on that logical record is interpreted as a comment and ignored.

Comments

Both line mode and block mode methods of commenting are supported:

- An asterisk (*) or slash-asterisk (/*) in column-1 marks the entire line as a comment. This style of comment is not allowed inside a continued /* */ type comment but is allowed in a continued token/value statement. A line that is entirely blank can also be considered a comment.
- Entire lines, blocks of lines, or portions of a line may be commented by beginning the comment with a slash-asterisk (/*) and terminating the comment with an asterisk-slash (*). Nested comments are honored.

About DB2 Cloning Tool Table Space Cloning token errors

Many tokens in CKZINI have default values. If a token has a default value and there is an error in the token, DB2 Cloning Tool Table Space Cloning substitutes the default value. DB2 Cloning Tool Table Space Cloning then outputs a message describing the error and the default value substitution and produces a return code of four (4).

- If MAX_RC=4, DB2 Cloning Tool Table Space Cloning continues to run to completion.
- If MAX_RC=0, DB2 Cloning Tool Table Space Cloning stops after parameter validation.

CKZINI keyword syntax and descriptions

The tables in this topic show the syntax and describe the keywords used in the many sections in the CKZINI member. Read through this topic to locate the keywords you want to understand or modify.

:PRODUCT_INFO section

Table 68. Keywords and values for the :PRODUCT_INFO section

Keywords and Values	Description and Usage
DB2_CLONING_TOOL_REL = <i>version/release number</i>	Used to verify product version and release. Do not alter.
DB2_CLONING_TOOL_REL_DATE = <i>release date</i>	Used to verify product release date. Do not alter.

:INIMERGE_VALUES section

Important: This section is critical for the INIMERGE process and should not be changed except by authorized IBM Software Support personnel.

Table 69. Keywords and values for the :INIMERGE_VALUES section

Keywords and Values	Description and Usage
SPECIAL_SECTIONS =	<p>The value for this token will be updated only by IBM Software. For Version V3.1, the values are: SPECIAL_SECTIONS=SI040_VALUES SI027_VALUES</p> <ul style="list-style-type: none"> • :SI027_VALUES Changes or additions to this section must be authorized by and under the direction of IBM Software Support. • :SI040_VALUES Changes or additions to this section must be authorized by and under the direction of IBM Software Support.

DB2 Cloning Tool Subsystem Cloning sections

The following sections are used for DB2 Cloning Tool Subsystem Cloning.

:DB2_CLONING_TOOL_OPTIONS section

Table 70. Keywords and values for the :DB2_CLONING_TOOL_OPTIONS section

Keywords and Values	Description and Usage
SPACE_MANAGEMENT = <i>option</i>	<p>Valid options are: HSM, DMS, ABR(#), or NONE. HSM, DMS, and ABR(#) can be specified as a single option, or, together. NONE may not be specified with any other option.</p> <p>HSM and ABR(#) indicate that DB2 Cloning Tool should use a volume serial of "MIGRAT" as an indication that a data set has been migrated.</p> <p>DMS indicates that DB2 Cloning Tool should use a volume serial of "ARCIVE" as an indication that a data set has been migrated.</p> <p>The default is HSM.</p>
CONCURRENT_EXECUTIONS = Y <u>N</u>	<p>This option addresses the situation where multiple jobs are running concurrently and need exclusive control over the same BCS.</p> <p>An N indicates that if another job has exclusive control over a BCS, the DB2 Cloning Tool job that wants exclusive control of the same BCS will fail with an error.</p> <p>The Y indicates that if another job has exclusive control over a BCS, the DB2 Cloning Tool job that wants exclusive control of the same BCS will wait until the BCS is available. The maximum time to wait is specified with the CONCURRENT_EXECUTIONS_WAIT_TIME keyword.</p> <p>The default is N.</p>
CONCURRENT_EXECUTIONS_WAIT_TIME = <i>nnn</i> <u>5</u>	<p>Specifies the maximum wait time in minutes to be used if CONCURRENT_EXECUTIONS = Y is specified. The maximum value that can be specified is 999.</p> <p>The default is 5.</p>

:COPY_OPTIONS section

Table 71. Keywords and values for the :COPY_OPTIONS section

Keyword and Values	Description and Usage
<p>CATWORK_ATTR= <i>catalog work data set allocation attributes</i></p>	<p>Catalog "work" data sets contain catalog entries captured during the COPY step and passed to other steps. This token controls allocation attributes for these data sets if not specified by the COPY CKZIN control statements.</p> <p>Specify attributes in TSO ALLOCATE syntax, e.g., UNIT(SYSALLDA) SPACE(10 10) CYLINDERS.</p> <p>The attributes that can be specified are:</p> <ul style="list-style-type: none"> • DATACLAS(data class name) • MGMTCLAS(management class name) • SPACE(quantity increment) • STORCLAS(storage class name) • TRACKS/CYLINDERS • UNIT(unit) • VOLUME(serial)
<p>TARGET_VOLS_SHOULD_BE_EMPTY = Y <u>N</u></p>	<p>Performs a check during the volume pairing process to ensure the target volumes are empty before issuing FlashCopy or SnapShot. Consider the following items:</p> <ul style="list-style-type: none"> • In the event a subsequent RENAME fails and the COPY must be rerun, DB2 Cloning Tool will not clean off the target volumes if "Y" was specified for this parameter. Either initialize the target volumes or change this keyword to "N". • If the "eliminated" target volumes cause there to be more source volumes than targets, the COPY will fail. • If the "eliminated" target volumes still leave at least as many target volumes as source volumes, the pairing will continue as usual. <p>The default is N.</p>

:DB2_CLONING_TOOL_DEFAULTS section

Table 72. Keywords and values for the :DB2_CLONING_TOOL_DEFAULTS section

Keywords and Values	Description and Usage
GDG_ALL_MIGRATED = <u>SKIP</u> RETAIN	<p>This option addresses the situation where a GDG matches a RENAME mask and all of the generations have been migrated.</p> <p>The GDG may be skipped, or the GDS entry may be retained with its new target name.</p> <ul style="list-style-type: none"> • SKIP = BCS update should skip the entry. • RETAIN = the migrated entries should be copied as is to the target BCS. <p>Note: If RETAIN is used, because the migrated generations do not exist under the new name, subsequent access to any generation will fail whether it is accessed specifically or via specification of the base name only. This option is provided to retain relativity.</p> <p>Important Note: To avoid destroying the relativity of active generations, DB2 Cloning Tool does NOT allow removing selected generations. For data that is migrated and is required on the target volumes, they must be recalled prior to the COPY.</p> <p>The default is SKIP.</p>
GDG_ALL_MIGRATED_RETAIN_RC = blank 0 4	Specifies the return code to be used if GDG_ALL_MIGRATED = RETAIN is specified.
GDG_EMPTY = <u>SKIP</u> RETAIN	<p>This option addresses an empty base GDG that matches a RENAME mask.</p> <p>The GDG entry can be skipped, or the new base entry can be added to the target user catalog.</p> <p>The default is SKIP.</p>
GDG_EMPTY_RETAIN_RC = blank 0 4	Specifies the return code to be used if GDG_EMPTY = RETAIN is specified.

Table 72. Keywords and values for the :DB2_CLONING_TOOL_DEFAULTS section (continued)

Keywords and Values	Description and Usage
GDG_MIGRATED = <u>ERROR</u> RETAIN	<p>This option addresses the situation where a GDG matches a rename mask and at least one generation is indeed found on a volume, yet one or more generations are migrated.</p> <p>The migrated generation may be treated as an ERROR, or the GDS entry in the GDG base record may be RETAINED with a corresponding return code of 0 or 4.</p> <ul style="list-style-type: none"> • ERROR - BCS update should terminate. • RETAIN - The migrated entry should be copied as is to the target BCS. <p>Note: If RETAINED, because the migrated generation does not exist under the new name, subsequent access to the generation will fail whether it is accessed specifically or via specification of the base name only.</p> <p>To avoid destroying the relativity of active generations, DB2 Cloning Tool does not allow removing selected generations.</p> <p>Retaining non-existent migrated generations may be suitable for situations such as overstated GDG limits (where it is normal for older generations to be migrated and hopefully never accessed), Log Files, etc. where perhaps only the current generation is kept on primary and older migrated generations are kept as a safety factor.</p> <p>The default is ERROR.</p>
GDG_MIGRATED_RETAIN_RC = blank 0 4	Specifies the return code to be used if GDG_MIGRATED = RETAIN is specified.

Table 72. Keywords and values for the :DB2_CLONING_TOOL_DEFAULTS section (continued)

Keywords and Values	Description and Usage
GDG_TAPE = <u>ERROR</u> RETAIN	<p>This option addresses the situation where a GDG matches a RENAME mask and at least one generation is indeed found on a volume, yet one or more generations are on tape.</p> <p>The tape generation may be treated as an ERROR, or the GDS entry in the GDG base record may be RETAINED with a corresponding return code of 0 or 4.</p> <ul style="list-style-type: none"> • ERROR - BCS update should terminate. • RETAIN - The tape entry should be copied as is to the target BCS. <p>Note: If RETAIN is specified, accessing a target tape GDS will cause a S813 ABEND whether it is accessed specifically or via specification of the base name only.</p> <p>To avoid destroying the relativity of active generations, DB2 Cloning Tool does not allow removing selected generations.</p> <p>Retaining non-existent tape generations may be suitable for situations such as overstated GDG limits where older generation may have been created on tape.</p> <p>The default is ERROR.</p>
GDG_TAPE_RETAIN_RC = blank 0 4	<p>Specifies the return code to be used if GDG_TAPE = RETAIN is specified.</p>
ISSUE_CKZ14141I = option	<p>Issues the message CKZ14141I when a data set matches the RENAME-MASKS, but was not on the DB2 Cloning Tool source volumes.</p> <p>You can customize which data set the error reports on using the option(s) specified in the CKZINI member. The DASD, MIG, and TAPE options may be specified in any combination.</p> <ul style="list-style-type: none"> • <u>ALL</u> - Issue message for any data set that matches the RENAME-MASKS, but is not on the source volumes. • DASD - Issue message for any DASD data set that matches the RENAME-MASKS, but is not on the source volumes. • MIG - Issue message for any migrated data set that matches the RENAME-MASKS. • TAPE - Issue message for any tape data set that matches the RENAME-MASKS. • NOMSG - Do not issue message CKZ14141I. <p>The default is ALL.</p>

Table 72. Keywords and values for the :DB2_CLONING_TOOL_DEFAULTS section (continued)

Keywords and Values	Description and Usage
MAX_RENAME_TASKS = <i>mmm</i>	<p>The <i>mmm</i> value specifies the maximum number of subtasks used by the RENAME command for volume processing if the RENAME CKZIN MAXTASKS is not supplied. At some point, increasing the number of subtasks will cease to increase performance, due to resource contention. Specifying a value that is too large may result in termination due to memory constraints. The maximum allowed value is 255.</p> <p>The default is 5.</p>
MISSING_USERCAT_DISP = DELETE <u>KEEP</u>	<p>Specifies the disposition of target volume data sets where the VVDS catalog back-pointer is not a catalog in the list supplied to the COPY step. This value is used if the corresponding keyword is not specified in the RENAME CKZIN control statements.</p> <p>The default is KEEP.</p>
MISSING_USERCAT_RC = 0 <u>4</u> 8	<p>Specifies the return code to be generated for the RENAME command if one or more target volume data sets contain a VVDS catalog back-pointer not in the list supplied to the COPY step. This value is used if the corresponding keyword is not specified in the RENAME CKZIN control statements.</p> <p>The default is 4.</p>
NOT_RENAMED_DISP = DELETE <u>KEEP</u>	<p>Specifies the disposition of target volume data sets not-renamed because they do not match a rename mask. This value is used if the corresponding keyword is not specified in the RENAME CKZIN control statements.</p> <p>The default is KEEP.</p>
NOT_RENAMED_RC = 0 4 <u>8</u>	<p>Specifies the return code to be generated for the RENAME command if one or more target volume data sets are not-renamed because they do not match a rename mask. This value is used if the corresponding keyword is not specified in the RENAME CKZIN control statements.</p> <p>The default is 8.</p>
ORPHAN_CATENTRY_DISP = DELETE <u>KEEP</u>	<p>Specifies the disposition of target volume data set catalog entries where in some circumstances the data set is not found on the volume. This value is used if the corresponding keyword is not specified in the RENAME CKZIN control statements.</p> <p>The default is KEEP.</p>

Table 72. Keywords and values for the :DB2_CLONING_TOOL_DEFAULTS section (continued)

Keywords and Values	Description and Usage
ORPHAN_CATENTRY_RC = 0 4 <u>8</u>	<p>Specifies the return code to be generated for the RENAME command if one or more target volume data set catalog entries do not have a corresponding volume data set. This value is used if the corresponding keyword is not specified in the RENAME CKZIN control statements.</p> <p>The default is 8.</p>
RECATALOG = Y <u>N</u>	<p>Specifies, if "Y", that catalog entries may be replaced if encountered when cataloging target volume data sets. This value is used if the corresponding keyword is not specified in the RENAME CKZIN control statements.</p> <p>The default is N.</p>
TEMP_DATASET_DISP = <u>DELETE</u> KEEP	<p>Specifies the disposition of temporary data sets found on target volumes. This value is used if the corresponding keyword is not specified in the RENAME CKZIN control statements.</p> <p>The default is DELETE.</p>
TEMP_DATASET_RC = 0 <u>4</u> 8	<p>Specifies the return code to be generated for the RENAME command if one or more temporary data sets are found on target volumes. This value is used if the corresponding keyword is not specified in the RENAME CKZIN control statements.</p> <p>The default is 4.</p>
VTOCIX_REBUILDER = <u>MSC</u> /* (MSC OR IBM) */	<p>Specifies the method for rebuilding the VTOCIX during the RENAME command.</p> <ul style="list-style-type: none"> • IBM - ICKDSF will be used to rebuild the VTOCIX. • MSC- The "on-board" VTOCIX rebuild will be used. <p>For Extended Address Volumes, ICKDSF will always be used to rebuild the VTOCIX.</p> <p>The default is MSC.</p>

Table 72. Keywords and values for the :DB2_CLONING_TOOL_DEFAULTS section (continued)

Keywords and Values	Description and Usage
RENAME_ERROR = <u>ABORT</u> CONTINUE	<p>This option specifies how processing proceeds when a RENAME error is encountered. This value is used if the corresponding keyword is not specified in the RENAME CKZIN control statements.</p> <p>ABORT will terminate with an RC=8 after the first error to preserve integrity. ABORT is recommended.</p> <p>CONTINUE will continue processing after most errors and the RENAME command will complete with the specified return code unless an error not handled by the CONTINUE logic is encountered.</p> <p>WARNING: The use of CONTINUE can cause inconsistencies between the contents of the volumes and catalogs. Possible problems include:</p> <ul style="list-style-type: none"> • Data sets could be cataloged but are not renamed on disk. • Data sets could be renamed on disk but are not cataloged. • Data sets that are not renamed on disk may not be deleted from disk. • GDG base and GDS entries will not exist in the catalog when there is a missing GDS. • A catalog entry may not point at the correct volume, a catalog entry may be invalid. • CONTINUE could leave uncataloged data sets on SMS managed volumes. <p>If this keyword is specified, DB2 Cloning Tool will not guarantee integrity and the given results will not be fixed by DB2 Cloning Tool.</p> <p>The default is ABORT.</p>
RENAME_ERROR_ CONTINUE_RC = 0 4 <u>8</u>	<p>Specifies the return code to be used if RENAME_ERROR = CONTINUE is specified. This value is used if the corresponding keyword is not specified in the RENAME CKZIN control statements.</p> <p>The default is 8.</p>

Table 72. Keywords and values for the :DB2_CLONING_TOOL_DEFAULTS section (continued)

Keywords and Values	Description and Usage
ISSUE_VCLOUSE = <u>YES</u> NO BEFORE AFTER	<p>Specifies if a catalog modify command will be issued as part of the volume RENAME processing. The catalog modify command is: F CATALOG,VCLOUSE(<i>targetvolser</i>)</p> <p>The Catalog address space (CAS), caches VVDS information. The modify command requests that the VVDS information cached for the target volume be refreshed. This value is used if the corresponding keyword is not specified in the RENAME CKZIN control statements.</p> <p>Possible values for this keyword are:</p> <ul style="list-style-type: none"> • NO specifies the modify command will NOT be issued. • BEFORE specifies that the modify command will be issued only before the VVDS is updated. • AFTER specifies that the modify command will be issued only after the VVDS has been updated. • YES specifies the modify command will be issued both before the VVDS is updated and after the VVDS has been updated. <p>The default is YES.</p>
ISSUE_VCLOUSE_SCOPE = <u>LOCAL</u> SYSPLEX	<p>If the CKZINI parameter ISSUE_VCLOUSE = YES NO BEFORE AFTER is set to YES, BEFORE, OR AFTER, use one of the following values for the ISSUE_VCLOUSE_SCOPE keyword:</p> <ul style="list-style-type: none"> • LOCAL – The catalog modify command, F CATALOG,VCLOUSE(<i>targetvolser</i>) will be issued only on the system that RENAME is running on. • SYSPLEX – The catalog modify command, F CATALOG,VCLOUSE(<i>targetvolser</i>), will be issued on the local system, and the modify command will be routed to all the other systems in the Sysplex, via an MVS ROUTE *OTHER command, after the VVDS has been updated. <p>This value is used if the corresponding keyword is not specified in the RENAME CKZIN control statements.</p> <p>The default is LOCAL.</p>

:DB2_OPTIONS section

Table 73. Keywords and values for the :DB2_OPTIONS section

Keyword or Value	Description
DB2_XCFCLEAN = <u>Y</u> N Restrictions: This parameter is only used if DB2 data sharing is being used.	Specifies that DB2UPDATE should clean up the target DB2 data-sharing group XCF structures and group members. This value is used if the corresponding keyword is not specified in the DB2UPDATE CKZIN control statements. The default is Y.
DB2_PLAN = <i>planname</i>	Specifies the plan name that will be used to process SQL statements with DB2. The default is CKZPLAN.

:RESOURCE_SERIALIZATION section

Installations running CA-MIM/MII with multiple systems and shared DASD need to set the following parameter to "YES" to ensure that when CA-MIM/MII GDIF is inactive, the DB2 Cloning Tool data sets are protected from data corruption.

Table 74. Keywords and values for the :RESOURCE_SERIALIZATION section

Keyword or Value	Description
MIM_GDIF = <u>NO</u> YES	If you have CA-MIM/MII with multiple systems and shared DASD, this token should be changed to YES to ensure that when CA-MIM/MII GDIF is inactive, the DB2 Cloning Tool data sets are protected from data corruption. The default is NO.

DB2 Cloning Tool Table Space Cloning sections

The following sections are used for DB2 Cloning Tool Table Space Cloning.

:DSN_PRODUCT_PERF section

Table 75. Keywords and values for the :DSN_PRODUCT_PERF section

Keyword or Value	Description
MAX_COPY_RC = <u>0</u> 4 8	STOP JOB WHEN > MAX_COPY_RC OCCURS WHEN COPYING DATA. This parm only applies to copy processing return codes. This allows one or more copies to fail and the others to continue. The default is 0.

Table 75. Keywords and values for the :DSN_PRODUCT_PERF section (continued)

Keyword or Value	Description
MAX_RC = <u>0</u> 4	<p>STOP JOB WHEN > MAX_RC</p> <p>This is the maximum value a return code from a DB2 Cloning Tool Table Space Cloning process may be and the job allowed to continue to run. This excludes data set copy processing (see MAX_COPY_RC).</p> <p>If some table spaces or index spaces are not found on the target and this parm is set to zero (0), no table spaces or index spaces will be copied. Set MAX_RC to four (4) if you want to complete the cloning process for table spaces and index spaces that can be copied, even if the remainder of the table spaces and index spaces cannot be cloned.</p> <p>Example:</p> <p>To stop a job on any warning message, set MAX_RC=0. To allow DB2 Cloning Tool Table Space Cloning to complete with one or more warning messages, set MAX_RC=4.</p> <p>The default is 0.</p>
MAX_SUBTASKS = <u>1</u> ... 99	<p>Valid values range from 1 to 99.</p> <p>DB2 Cloning Tool Table Space Cloning uses subtasks to perform several functions, such as catalog access in the source job and SYNCDB2 command processing in the target job. These subtasks allow multiple IOs to be performed concurrently. Changing the number of subtasks DB2 Cloning Tool Table Space Cloning uses may improve performance.</p> <p>Raising the limit of the number of subtasks that can be specified may reduce the elapsed time of the target job. At some point, the larger number of subtasks will place a burden on z/OS and the elapsed time will start to go back up. Each different set of objects may have its own optimal target job value for the number of subtasks.</p> <p>The larger number of subtasks available normally is useful only for the target job. When using more than 18 subtasks in the target job, use a PARMLIB member for the source job and another for the target job. From one to four subtasks is generally sufficient for the source job. The TCPIP server job uses only one subtask, regardless of how many are specified. If the SET command is used, the source and target jobs get the same number of subtasks. This SET value also overrides the value in PARMLIB.</p> <p>The default is 1.</p>

:TCPIP_OPTIONS section

Table 76. Keywords and values for the :TCPIP_OPTIONS section

Keyword or Value	Description
TCPIP_SERVER_PORT = <u>5099</u>	<p>TCPIP SERVER PORT #</p> <p>This is the port that the DB2 Cloning Tool Table Space Cloning TCP/IP server job listens on for client connects. It must be known to the source job and the TCP/IP server job and must be the same value. It should be an unused number less than 65536.</p> <p>One TCP/IP server only connects to a single DB2 subsystem. If multiple servers are required on a single LPAR image, use a different port number for each server.</p> <p>The default port for this product is 5099.</p>
TCPIP_STC_NAME = <u>TCPIP</u>	<p>LOCAL TCPIP STC NAME</p> <p>This defines the name of the TCP/IP started task.</p> <p>TCPIP is the default used by this product, as it is the MVS default.</p>

:DSN_COPY_OPTIONS section

This section defines the options to be used for the data set copies. These may be overridden on the COPY command for the target.

Table 77. Keywords and values for the :DSN_COPY_OPTIONS section

Keyword or Value	Description
ALWAYS_COPY_INDEXSPACES = <u>Y</u> <u>N</u>	<p>This token determines how index spaces are included in LISTDEFS. When the value is set to Y, all index spaces are included for every table space included in a LISTDEF. No INCLUDE INDEXSPACES syntax is required in the LISTDEF.</p> <p>The default is N.</p>
AUTO_START_SOURCE_SPACE = <u>Y</u> <u>N</u> <u>R</u>	<p>This token allows the source job to optionally start each source DB2 table spaces and index spaces after the copy process is complete.</p> <p>When the value is Y, DB2 Cloning Tool Table Space Cloning starts the source table spaces and index spaces after the copy is complete.</p> <p>When the value is N, source spaces are left stopped after the copy is complete.</p> <p>When the value is R, the source table spaces and index spaces are restored to the status they were before executing the source job.</p> <p>If you set this parm to Y or R, and a table space or index space has a status that is not STOP, RW, RO or STOPP, a RC of eight (8) is issued.</p> <p>The default is Y.</p>

Table 77. Keywords and values for the :DSN_COPY_OPTIONS section (continued)

Keyword or Value	Description
AUTO_START_TARGET_SPACE = <u>Y</u> N	<p>DB2 Cloning Tool Table Space Cloning will issue a START command for the target space.</p> <p>This token determines if SYNCDB2 processing in the target job issues a DB2 START DATABASE SPACENAM command for the table space or index space being processed after all the Sync IO to the data set is complete. This value determines the value of the START-SPACE command in each invocation of the SYNCDB2 command in the target job.</p> <p>Note: There is one SYNCDB2 command for each target data set in the SYNCDB2 command member.</p> <p>When DB2 Cloning Tool Table Space Cloning builds the SYNCDB2 commands, it uses the value of AUTO-START-TARGET-SPACE from PARMLIB and/or the COPY command to add a similar command to the SYNCDB2 command output member. Note that the COPY command value overrides PARMLIB.</p> <p>The default is Y.</p>
AUTO_STOP_TARGET_SPACE = <u>Y</u> N	<p>This parm determines if DB2 Cloning Tool Table Space Cloning issues a STOP DATABASE SPACENAM DB2 command for each target table space and index space. If the value is Y, DB2 Cloning Tool Table Space Cloning will issue a STOP command before the copy begins.</p> <p>Note: If a STOP is issued, it must complete before the copy can begin as DB2 Cloning Tool Table Space Cloning must have exclusive control of the target data set.</p> <p>If the value is N, DB2 Cloning Tool Table Space Cloning assumes the space is already stopped and no DB2 command is issued before the copy begins. If the space is not stopped, an allocation error is issued.</p> <p>The default is Y.</p>
COPY_IF_NO_DB2_TARGET_OBJECTS = Y <u>N</u>	<p>Copy even if no DB2 table spaces or index spaces exist on the target.</p> <p>This parm enables DB2 Cloning Tool Table Space Cloning to copy data sets when no target table spaces or index spaces exist. DB2 Cloning Tool Table Space Cloning will use defaults for the high level qualifier (DEFVCAT subcommand from the COPY command) and the fifth level qualifier (F0001) when creating the target data set names.</p> <p>The default is N, which means do not copy the source table spaces or index spaces if no target table spaces or index spaces exist.</p>
DSNS_PER_COPY = 255	<p>The number of data sets to include in each DSS copy command. Valid range: 1 to 255. A performance benefit may be realized by changing this value.</p>

Table 77. Keywords and values for the :DSN_COPY_OPTIONS section (continued)

Keyword or Value	Description
DSS_COPY_COMMANDS = 24	The number of DSS copy commands in each DSS call. Valid range: 1 to 256. A performance benefit may be realized by changing this value. Specifying a large value may result in a storage shortage and S878 abends. When cloning a large number of data sets, the default of 24 might be too high.
REPLACE_TARGET_DSN = <u>Y</u> N	<p>Replace the data set on the target if it exists.</p> <p>When the value is Y, this parm allows DB2 Cloning Tool Table Space Cloning to overlay the target data set with the source data set.</p> <p>When the value is N, DB2 Cloning Tool Table Space Cloning will create another data set with the same name as the target data set except for the fifth level qualifier.</p> <p>Note that the target fifth level qualifier will be obtained from the target catalog for the table space or index space being copied. If there is no target table space or index space and parms allow the source data set to be copied if no DB2 table space or index space exists, F0001 is used.</p> <p>The default is Y.</p>
RESET_LOGRBA = <u>Y</u> N	<p>Reset the LOGRBA in the target table space and index space data sets.</p> <p>This parm determines if SYNCDB2 processing in the target job resets the LOGRBA in the copied table space and index space data sets. This value determines the value of the RESET_LOGRBA command in each invocation of the SYNCDB2 command in the target job.</p> <p>Note there is one SYNCDB2 command for each target data set in the SYNCDB2 command member. Failure to specify Y may result in the DB2 table spaces and index spaces being unusable after completion of the target job.</p> <p>The default is Y.</p>

Deallocating target DB2 coupling facility structures

In a data-sharing environment, DB2 retains lock information and table information in coupling facility (CF) structures. These structures may exist from a previous running of the target DB2 sharing systems. The target DB2 will use this out-of-date information if it exists during start-up and will end up corrupting tables on the target DB2 subsystem. To prevent this, the target DB2 CF structures need to be deallocated prior to starting the target DB2.

The following steps describe how to determine if there any target DB2 CF structures that need to be deallocated and how to deallocate them.

1. For data sharing, ensure the target DB2 CF structures are deallocated.

The DB2 structure names are:

groupname_LOCK1
groupname_SCA
groupname_GBPxxx

Where:

- *groupname* is replaced with the name of the DB2 data-sharing group.
- *GBPxxxx* is one of these unique identifiers of the buffer pool:

GBP0, GBP1, ... GBP49 or
GBP8K0, ... GBP8K9 or
GBP16K0, ... GBP16K9 or
GBP32K, ... GBP32K9

2. Issue this z/OS command: **D XCF,STR**

3. Look for any target DB2 structures that are marked as "ALLOCATED".

- Partial output from D XCF,STR: (*groupname* = DBGT)

```
DBGT_GBP0          --          -- NOT ALLOCATED
DBGT_LOCK1         09/17/2009 10:14:29 ALLOCATED
DBGT_SCA           09/17/2009 10:14:27 ALLOCATED
```

- If *groupname_LOCK1* is marked as "ALLOCATED", issue the following z/OS commands:

```
SETXCF FORCE,CONNECTION,STRNAME=groupname_LOCK1,CONNAME=ALL
SETXCF FORCE,STR,STRNAME=groupname_LOCK1
```

- If *groupname_SCA* is marked as "ALLOCATED", issue the following z/OS command:

```
SETXCF FORCE,STR,STRNAME=groupname_SCA
```

- If any *groupname_GBPxxxx* is marked "ALLOCATED", issue the following z/OS commands:

```
SETXCF
FORCE,CONNECTION,STRNAME=groupname_GBPxxxx,CONNAME=ALL
SETXCF FORCE,STR,STRNAME=groupname_GBPxxxx
```

4. The following command should be issued before starting the target DB2 subsystem: **D XCF,STR**

The command response lines corresponding to the desired DB2 group should show as "NOT ALLOCATED". As an example, you are looking for *groupname* = x where x in this example is the DB2 data-sharing group named DBGT. This example is a partial output from the command of only the lines for the DB2 data-sharing group (*groupname*) named DBGT.

```
DBGT_GBP0          --          -- NOT ALLOCATED
DBGT_LOCK1         --          -- NOT ALLOCATED
DBGT_SCA           --          -- NOT ALLOCATED
```

Troubleshooting

Use these topics to diagnose and correct problems that you experience with DB2 Cloning Tool.

Messages

Naming conventions

Product message identifiers are in the form **pppmmmmnnx**

where:

ppp is the 3 alpha character product code – CKZ for DB2 Cloning Tool

mmm is the module identifier

nn is the message number and
 x is the message type indicator
 E error
 I information
 W warning

Example 1:

The message identifier **CKZ02056I** would be a message from module **CKZ00020**, message number of **56**, and is an Information only message.

Example 2:

The message identifier **CKZVSE11E** would be a message from module **CKZ01VSE**, message number of **11**, and is an Error type message.

Messages and descriptions**Tools Customizer messages:**

Use the information in these messages to help you diagnose and solve Tools Customizer problems.

CCQB000I The product parameter data was saved in the data store.

Explanation: Changes that were made to the product parameters were saved in the data store.

System action: None.

User response: No action is required.

CCQB001I The DB2 parameter data was saved in the data store.

Explanation: Changes that were made to the DB2 parameters were saved in the data store.

System action: None.

User response: No action is required.

CCQB002I The LPAR parameter data was saved in the data store.

Explanation: Changes that were made to the LPAR parameters were saved in the data store.

System action: None.

User response: No action is required.

CCQB003E At least one step must be selected in a selected task. The selected task is *task_description*.

Explanation: When a task is selected, at least one step must be selected. A selected step is missing from the specified task.

System action: Processing stops.

User response: Select a step in the specified task or deselect the task.

CCQB004I The required information to run the Discover EXEC was saved in the data store.

Explanation: The data store contains all the information that is required to run the Discover EXEC.

System action: None.

User response: No action is required.

CCQB005E The conflicting values for the *parameter_name* parameter must be resolved before the information can be saved.

Explanation: Two values for one parameter conflict with each other, and they must be resolved to save the information.

System action: Processing stops.

User response: Resolve the conflicting values for the parameter.

CCQB006E One row must be selected.

Explanation: One row in the table must be selected.

System action: Processing stops.

User response: Select one row.

CCQB007E Only one row can be selected.

Explanation: Multiple rows in the table are selected, but only one row is allowed to be selected.

System action: Processing stops.

User response: Select only one row.

CCQC000I The jobs have been customized on the selected DB2 entries.

Explanation: The jobs were customized on the DB2 entries that were selected.

System action: None.

User response: Press Enter to clear the message.

CCQC001W The jobs were not generated on one or more of the selected DB2 entries. Press PF3 to check the DB2 entries that were not customized.

Explanation: The product was not customized on one or more of the DB2 entries that were selected.

System action: None.

User response: Press PF3 to see the DB2 entries on which the product was not customized. The status of these DB2 entries is Errors in Customization.

CCQC002I The edit session was started automatically because values for required parameters are missing or must be verified.

Explanation: If product, LPAR parameters, or DB2 parameters are not defined or if parameter definitions must be verified, an editing session for the undefined or unverified parameters starts automatically.

System action: None.

User response: Define values for all required product, LPAR parameters, or DB2 parameters.

CCQC003W The *template_name* template in the *library_name* metadata library does not contain any parameters.

Explanation: The specified template does not have parameters.

System action: None.

User response: No action is required.

CCQC004S The value of the "type" attribute for the *template_name* template in the *library_name* metadata library does not match the value that was previously specified. The value is *value_name*, and the previously specified value is *value_name*.

Explanation: The value of the "type" attribute must match the value that was previously specified.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQC005S The *template_name* template exceeds the number of allowed templates for a customization sequence. The template is in the *library_name* metadata library.

Explanation: The customization sequence can process only *number* templates. The specified template cannot be processed because the customization sequence already contains the maximum number of templates.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQC006E The jobs could not be generated for the *group_attach_name* DB2 group attach name.

Explanation: The customization jobs could not be generated for the specified DB2 group attach name.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQC007E The jobs could not be generated for the *subsystem_ID* DB2 subsystem.

Explanation: The customization jobs could not be generated for the specified DB2 subsystem.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQC008E The jobs could not be generated for the *member_name* DB2 member.

Explanation: The customization jobs could not be generated for the specified DB2 member.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQC009S The jobs were not generated for the DB2 entries.

Explanation: One or more errors occurred while customization jobs were being generated for the selected DB2 entries.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQC010S The *template_name* template could not be accessed in the *library_name* metadata library.

Explanation: The specified template could not be accessed because the user does not have RACF access to the data set, the data set has incorrect data characteristics, or the data set is not cataloged.

System action: Processing stops.

User response: Ensure that you have RACF access to the data set, that the characteristics are correct according to the specifications of the product that you are customizing, and that the data set is cataloged. If the problem persists, contact IBM Software Support.

CCQC011S The *template_name* template could not be written to the *library_name* customization library.

Explanation: The specified template could not be accessed because the user does not have RACF access to the data set, the data set has incorrect data characteristics, or the data set is not cataloged.

System action: Processing stops.

User response: Ensure that you have RACF access to the data set, that the characteristics are correct according to the specifications of the product that you are customizing, and that the data set is cataloged. If the problem persists, contact IBM Software Support.

CCQC012W The job card was generated with default values because the JOB keyword was missing.

Explanation: Default values were used to generate the job card because the JOB keyword was not specified in the first line of the job card.

System action: The job card was generated with default values.

User response: No action is required. To generate the job card with your own values, add the JOB keyword in the first line of the job card.

CCQC013W The job card was generated with the default value for the programmer name because the specified programmer name exceeded 20 characters.

Explanation: Default values were used to generate the job card because the specified programmer name contained too many characters.

System action: The job card was generated with default values.

User response: No action is required. To generate the job card with your own values, add a valid

programmer name in the job card. A valid programmer name is 1 - 20 characters.

CCQC014W The job card was generated with default values because the JOB keyword was not followed by a space.

Explanation: Default values were used to generate the job card because a space did not follow the JOB keyword.

System action: The job card was generated with default values.

User response: No action is required. To generate the job card with your own values, add a space after the JOB keyword in the job card.

CCQC015S The *template_name* template in the *library_name* metadata library contains the following file-tailoring control statement: *statement_name*. This control statement is not valid in a *template_type* template.

Explanation: The *template_type* template cannot contain the specified type of file-tailoring control statement.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQC016S The)DOT file-tailoring control statement exceeded the number of allowed occurrences for the *template_name* template in the *library_name* metadata library.

Explanation: The)DOT file-tailoring control statement can occur only a limited number of times in the specified template.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQC017S The nested)DOT file-tailoring control statements exceeded the number of allowed occurrences in the *template_name* template in the *library_name* metadata library.

Explanation: Nested)DOT file-tailoring control statements can occur only *number* times.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQC018S The *template_name* template in the *library_name* metadata library is not valid because it does not contain any data.

Explanation: The specified template is missing required data.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQC019S The *template_name* template in the *library_name* metadata library is not valid because an)ENDDOT file-tailoring control statement is missing.

Explanation: A)ENDDOT file-tailoring control statement is required in the specified template.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQC021S The *template_name* template in the *library_name* metadata library is not valid because the template must start with the *parameter_name* job card parameter.

Explanation: The specified template must start with the specified job card parameter.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQC022S The parameters used in a)DOT file-tailoring control statement exceeded the number of allowed parameters in the *template_name* template. The template is in the *library_name* metadata library. The error occurs in)DOT section *section_number*.

Explanation: A)DOT file-tailoring control statement can contain only a limited number of parameters.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQC023S The)DOT file-tailoring control statement must include the *table-name* table name in the *template_name* template. The template is in the *library_name* metadata library. The error occurs in)DOT section *section_number*.

Explanation: The)DOT file-tailoring control statement is missing a required table name.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQC024S ISPF file tailoring failed for the *template_name* template in the *library_name* metadata library.

Explanation: An error occurred during ISPF file tailoring for the specified template.

System action: Processing stops.

User response: Review the Tools Customizer-generated trace data set and the ISPF file tailoring trace data set. To create an ISPF file tailoring trace data set, complete the following steps:

1. Run Tools Customizer until the error is about to occur.
2. Specify the ISPFITRC command, and press Enter.
3. Issue the Tools Customizer command that issues the error.
4. Specify the ISPFITRC command, and press Enter. The ISPF file tailoring trace data set is created. It adheres to the following naming convention: *TSO_ID*.ISPFITRC, where *TSO_ID* is the TSO user ID that is being used.

If the problem persists, gather the following information and contact IBM Software Support.

- A screen capture of the Tools Customizer error. Ensure that the complete error message is displayed by pressing PF1.
- The Tools Customizer trace data set. It adheres to the following naming convention: *TSO_ID*.CCQ.TRACE, where *TSO_ID* is the TSO user ID that is running Tools Customizer.
- The ISPF file tailoring trace data set.

CCQC025I Customized jobs do not exist because they have not been generated.

Explanation: The list of customized jobs cannot be displayed because the product has not been customized for any DB2 entries.

System action: None.

User response: Complete the steps to customize a product. Customized jobs are generated when all required product, LPAR parameters, and DB2 parameters are defined and at least one DB2 entry on which to customize the product has been selected.

CCQC026S The value of the "customized" attribute for the *parameter_name* parameter in the *library_name* metadata library template does not match the value that was previously specified. The value is *value_name*, and the previously specified value is *value_name*.

Explanation: The value for the "customized" attribute for a parameter must match the value that was previously specified.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQC027S The *job_name* customization job was not found in the *library_name* customization library.

Explanation: The selected customization job does not exist in the customization library.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQC028S The *library_name* customization library was not found.

Explanation: The customization library does not exist.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQC029I The customization jobs were generated for *Product_name*.

Explanation: The customization jobs were generated for the specific product.

System action: None.

User response: No action is required.

CCQC030S The customization jobs cannot be generated because at least one DB2 entry must be associated with this product.

Explanation: The product that you are customizing requires at least one DB2 entry to be associated with it before customization jobs can be generated.

System action: None.

User response: Associate a DB2 entry with the product that you are customizing, and regenerate the jobs.

CCQC031I The jobs were generated for the associated DB2 entries.

Explanation: The customization jobs were generated for the DB2 entries that are associated with the product.

System action: None.

User response: No action is required.

CCQC032S The customization jobs were not generated for *Product_name*.

Explanation: A severe error occurred while the jobs were being generated for the specified product.

System action: None.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQC033S The *customization_library_name* has no customized jobs.

Explanation: The specified customization library cannot be browsed or edited because it is empty.

System action: None.

User response: Generate customization jobs for the specified library, and browse or edit the library again.

CCQC034S The specified operation is not allowed.

Explanation: Issuing commands against customization jobs from the customization library from an ISPF browse or edit session that was started on the Finish Product Customization panel is restricted.

System action: None.

User response: To make changes to customization jobs, follow the steps for recustomization.

CCQC035E Before you generate customization jobs, edit the product parameters to select one or more tasks or steps, and then issue the G line command or the GENERATEALL command again.

Explanation: One or more tasks or steps must be selected before customization jobs can be generated.

System action: None.

User response: Edit the product parameters to select one or more tasks or steps. Then, issue the G line command or the GENERATEALL command again.

CCQC036E Before you exit the Product Parameters panel, you must select one or more tasks or steps to generate customization jobs or issue the CANCEL command.

Explanation: One or more tasks or steps must be selected to generate customization jobs or the CANCEL command must be issued before you can exit the Product Parameters panel.

System action: None.

User response: Select one or more tasks or steps, or issue the CANCEL command.

CCQD000W The *member_name* environment index member is not valid. The PL/I XML parser issued the following exception warning code: *code_number*.

Explanation: While determining if the specified environment index member is valid, the PL/I XML parser issued an exception warning code.

System action: Processing continues.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the warning.

CCQD001S The *member_name* environment index member is not valid. The PL/I XML parser issued the following exception error code: *code_number*.

Explanation: While determining if the specified environment index member is valid, the PL/I XML parser issued an exception error code.

System action: Processing continues.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the error.

CCQD002S The XML structure of the *member_name* environment index member is not valid. The *element_name* element is unknown.

Explanation: The specified environment index member contains an unknown element.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD003S The XML structure of the *member_name* environment index member is not valid. Content is not allowed for the *element_name* element, but content was found.

Explanation: Content was found in an element that cannot contain content.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD004S The XML structure of the *member_name* environment index member is not valid. Content is required for the *element_name* element, but content was not found.

Explanation: The specified element does not contain required content.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD005S The XML structure of the *member_name* environment index member is not valid. The content length for the *element_name* element exceeds *maximum_number* characters.

Explanation: The specified element contains too many characters.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD006S The XML structure of the *member_name* environment index member is not valid. The *element_name* element cannot occur more than *maximum_number* times.

Explanation: The specified element occurs too many times in the environment index member.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD007S The XML structure of the *member_name* environment index member is not valid. The *element_name* element must occur at least *minimum_number* times.

Explanation: The specified element does not occur enough times in the environment index member.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD008S The XML structure of the *member_name* environment index member is not valid. The *attribute_name* attribute in the *element_name* element cannot occur more than *maximum_number* times.

Explanation: The specified attribute occurs too many times in the environment index member.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD009S The XML structure of the *member_name* environment index member is not valid. The *attribute_name* attribute in the *element_name* element must occur at least *minimum_number* times.

Explanation: The specified attribute does not occur

enough times in the environment index member.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD010S The XML structure of the *member_name* environment index member is not valid. Content is not allowed for the *attribute_name* attribute in the *element_name* element, but content was found.

Explanation: Content was found in an attribute that cannot contain content. The name of the attribute and the name of the element that contains it are indicated in the message text.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD011S The XML structure of the *member_name* environment index member is not valid. Content is required for the *attribute_name* attribute in the *element_name* element, but content was not found.

Explanation: An attribute does not contain required content. The name of the attribute and the name of the element that contains it are indicated in the message text.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD012S The XML structure of the *member_name* environment index member is not valid. The content length for the *element_name* element exceeds *maximum_number* characters.

Explanation: An element contains too many characters. The name of the element and the maximum number of allowed characters are indicated in the message text.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD013S The XML structure of the *member_name* environment index member is not valid. The *attribute_name* attribute in the *element_name* element is unknown.

Explanation: The environment index member contains an unknown attribute. The name of the unknown

attribute and the name of the element that contains it are indicated in the message text.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD050S The following LPAR serial number is duplicated in the environment index member: *serial_number*.

Explanation: The environment index member contains duplicate LPAR serial numbers. The duplicate serial number is indicated in the message text.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD051S The following DB2 serial number is duplicated in the environment index member: *serial_number*.

Explanation: The environment index member contains duplicate DB2 serial numbers. The duplicate serial number is indicated in the message text.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD052S The following DB2 group attach name is duplicated in the environment index member: *group_attach_name*.

Explanation: The environment index member contains duplicate group attach names.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD053S The reference to the following DB2 subsystem for a DB2 group attach name is duplicated in the environment index member: *subsystem_ID*.

Explanation: The environment index member contains duplicate references to a DB2 subsystem for a DB2 group attach name.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD054S The reference to the following DB2 subsystem for the *LPAR_name* LPAR is duplicated in the environment index member: *subsystem_ID*.

Explanation: The environment index member contains duplicate references to a DB2 subsystem for an LPAR. The duplicate subsystem ID is indicated in the message text.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD055S The following DB2 group attach name was not found in the environment index member: *group_attach_name*.

Explanation: A group attach name that is referenced by a DB2 member does not exist in the environment index member.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD056S The following LPAR was not found in the environment index member: *LPAR_name*.

Explanation: The LPAR does not exist in the environment index member.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD057S The following LPAR is duplicated in the environment index member: *LPAR_name*.

Explanation: The environment index member contains duplicate LPARs. The name of the duplicate LPAR name is indicated in the message text.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD100W The *member_name* product index member is not valid. The PL/I XML parser issued the following exception warning code: *code_number*.

Explanation: While determining if the product index member is valid, the PL/I XML parser issued the specified exception warning code.

System action: Processing continues.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the

specified exception warning code.

CCQD101S The *member_name* product index member is not valid. The PL/I XML parser issued the following exception error code: *code_number*.

Explanation: While determining if the product index member is valid, the PL/I XML parser issued the specified exception error code.

System action: Processing stops.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the specified exception error code. Ensure that the Tools Customizer data store data set DCB is the same as the sample SCCQSAMP(CCQCDATS) data set DCB.

CCQD102S The XML structure of the *member_name* product index member is not valid. The *element_name* element is unknown.

Explanation: The specified product index member contains an unknown element.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD103S The XML structure of the *member_name* product index member is not valid. Content is not allowed for the *element_name* element, but content was found.

Explanation: Content was found for an element that cannot contain content.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD104S The XML structure of the *member_name* product index member is not valid. Content is required for the *element_name* element, but content was not found.

Explanation: The specified element does not contain required content.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD105S The XML structure of the *member_name* product index member is not valid. The content length for the *element_name* element exceeds *maximum_number* characters.

Explanation: The specified element contains too many characters.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD106S The XML structure of the *member_name* product index member is not valid. The *element_name* element cannot occur more than *maximum_number* times.

Explanation: The specified element occurs too many times in the product index member.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD107S The XML structure of the *member_name* product index member is not valid. The *element_name* element must occur at least *minimum_number* times.

Explanation: The specified element does not occur enough times in the product index member.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD108S The XML structure of the *member_name* product index member is not valid. The *attribute_name* attribute in the *element_name* element cannot occur more than *maximum_number* times.

Explanation: An attribute occurs too many times. The name of the attribute and the element that contains it are indicated in the message text.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD109S The XML structure of the *member_name* product index member is not valid. The *attribute_name* attribute in the *element_name* element must occur at least *minimum_number* times.

Explanation: The specified attribute does not occur enough times in the product index member.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD110S The XML structure of the *member_name* product index member is not valid. Content is not allowed for the *attribute_name* attribute in the *element_name* element, but content was found.

Explanation: An attribute cannot contain content. The name of the attribute and the element that contains it are indicated in the message text.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD111S The XML structure of the *member_name* product index member is not valid. Content is required for the *attribute_name* attribute in the *element_name* element, but content was not found.

Explanation: An attribute requires content. The name of the attribute and the name of the element that contains it are indicated in the message text.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD112S The XML structure of the *member_name* product index member is not valid. The content length for the *element_name* element exceeds *maximum_number* characters.

Explanation: The specified element contains too many characters.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD113S The XML structure of the *member_name* product index member is not valid. The *attribute_name* attribute in the *element_name* element is unknown.

Explanation: The specified attribute in the product index member is unknown.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD118S The content of the *member_name* product index member is not valid. The *configuration_ID* configuration ID for the *configuration-name* configuration name is not unique.

Explanation:

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD120S The content of the *member_name* product index member is not valid. The pack ID *pack_ID* that is referenced by product prefix *product_prefix* in the metadata library *library_name* could not be found.

Explanation: The specified pack ID could not be found in the metadata library.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD121I The specified pack contains the *component_name*, which was previously specified as a stand-alone product.

Explanation: The specified component of the pack was previously specified as a stand-alone product.

System action: None.

User response: No action is required.

CCQD122I The specified component metadata library was previously specified as part of the *pack_name*.

Explanation: The specified metadata library for the component was previously specified as part of a pack.

System action: None.

User response: No action is required.

CCQD123E The customization library name *library_name* is being used by another product or component. Specify another customization library qualifier on the Tools Customizer Settings panel.

Explanation: A different product or component is using the specified customization library.

System action: None.

User response: Specify another customization library qualifier on the Tools Customizer Settings panel.

CCQD300W The *member_name* product environment member is not valid. The PL/I XML parser issued the following exception warning code: *code_number*.

Explanation: While determining if the product environment member is valid, the PL/I XML parser issued the specified exception warning code.

System action: Processing continues.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the specified exception warning code.

CCQD301S The *member_name* product environment member is not valid. The PL/I XML parser issued the following exception error code: *code_number*.

Explanation: While determining if the product environment member is valid, the PL/I XML parser issued the specified exception error code.

System action: Processing stops.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the specified exception error code.

CCQD302S The XML structure of the *member_name* product environment member is not valid. The *element_name* element is unknown.

Explanation: The specified product environment member contains an unknown element.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD303S The XML structure of the *member_name* product environment member is not valid. Content is not allowed for the *element_name* element, but content was found.

Explanation: Content was found for an element that cannot contain content.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD304S The XML structure of the *member_name* product environment member is not valid. Content is required for the *element_name* element, but content was not found.

Explanation: The specified element does not contain required content.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD305S The XML structure of the *member_name* product environment member is not valid. The content length for the *element_name* element exceeds *maximum_number* characters.

Explanation: The specified element contains too many characters.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD306S The XML structure of the *member_name* product environment member is not valid. The *element_name* element cannot occur more than *maximum_number* times.

Explanation: The specified element occurs too many times in the product environment member.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD307S The XML structure of the *member_name* product environment member is not valid. The *element_name* element must occur at least *minimum_number* times.

Explanation: The specified element does not occur enough times in the product environment member.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD308S The XML structure of the *member_name* product environment member is not valid. The *attribute_name* attribute in the *element_name* element cannot occur more than *maximum_number* times.

Explanation: The specified attribute occurs too many times. The name of the attribute and the element that contains it are indicated in the message text.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD309S The XML structure of the *member_name* product environment member is not valid. The *attribute_name* attribute in the *element_name* element must occur at least *minimum_number* times.

Explanation: The specified attribute does not occur enough times in the product environment member.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD310S The XML structure of the *member_name* product environment member is not valid. Content is not allowed for the *attribute_name* attribute in the *element_name* element, but content was found.

Explanation: The specified attribute cannot contain content. The name of the attribute and the element that contains it are indicated in the message text.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD311S The XML structure of the *member_name* product environment member is not valid. Content is required for the *attribute_name* attribute in the *element_name* element, but content was not found.

Explanation: The specified attribute requires content. The name of the attribute and the name of the element that contains it are indicated in the message text.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD312S The XML structure of the *member_name* product environment member is not valid. The content length for the *element_name* element exceeds *maximum_number* characters.

Explanation: The specified element contains too many characters.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD313S The XML structure of the *member_name* product environment member is not valid. The *attribute_name* attribute in the *element_name* element is unknown.

Explanation: The specified attribute in the product environment member is unknown.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD350I The *subsystem_ID* DB2 subsystem is associated with this product.

Explanation: The specified DB2 subsystem was added and saved in the Tools Customizer data store for the product to be customized.

System action: Processing continues.

User response: No action is required.

CCQD351I The *member_name* DB2 member for the *group_attach_name* DB2 group attach name is associated with this product.

Explanation: The specified DB2 member for the group attach name was added and saved in the Tools Customizer data store for the product to be customized.

System action: Processing continues.

User response: No action is required.

CCQD352I The *group_attach_name* DB2 group attach name is associated with this product.

Explanation: The specified DB2 group attach name was added and saved in the Tools Customizer data store for the product to be customized.

System action: Processing continues.

User response: No action is required.

CCQD353E The *subsystem_ID* DB2 subsystem is already associated with this product.

Explanation: The specified DB2 subsystem cannot be added for the product to be customized because it already exists in the product environment in the data store.

System action: None.

User response: Ensure that the DB2 subsystem is specified correctly. If the problem persists, contact IBM Software Support.

CCQD354E The *member_name* DB2 member for the *group_attach_name* DB2 group attach name is already associated with this product.

Explanation: The specified DB2 member for the group attach name cannot be added for the product to be customized because it already exists in the product environment in the data store.

System action: None.

User response: Ensure that the DB2 group attach name is specified correctly. If the problem persists, contact IBM Software Support.

CCQD355E The *group_attach_name* DB2 group attach name is already associated with this product.

Explanation: The specified DB2 group attach name cannot be added for the product to be customized because it already exists in the product environment in the data store.

System action: Processing stops.

User response: Ensure that the DB2 group attach name is specified correctly. If the problem persists, contact IBM Software Support.

CCQD356S The *library_name* metadata library is already associated with the maximum number of allowed DB2 entries for this product.

Explanation: The specified metadata library cannot be associated with more DB2 entries because it is already associated with the number of DB2 entries that are allowed.

System action: Processing stops.

User response: Delete an associated DB2 entry, and associate the specified library with another DB2 entry again.

CCQD357I The *subsystem_ID* DB2 subsystem is unassociated with this product.

Explanation: The specified DB2 SSID was unassociated with the product that you are customizing.

System action: Processing continues.

User response: No action is required.

CCQD358I The *member_name* DB2 member for the *group_attach_name* DB2 group attach name is unassociated with this product.

Explanation: The specified DB2 member for the DB2

group attach name was unassociated with the product that you are customizing.

System action: Processing continues.

User response: No action is required.

CCQD359I The *group_attach_name* DB2 group attach name is unassociated with this product.

Explanation: The specified DB2 group attach name was unassociated with the product that you are customizing.

System action: Processing continues.

User response: No action is required.

CCQD360S The *library_name* metadata library is not associated with the specified DB2 subsystem *subsystem_ID*.

Explanation: The specified DB2 subsystem and metadata library are not associated with each other.

System action: None.

User response: Ensure that the DB2 subsystem and the metadata library are associated. If the problem persists, contact IBM Software Support.

CCQD361S The *library_name* metadata library is not associated with the specified DB2 data sharing group member *member_name* for the *group_attach_name* DB2 group attach name.

Explanation: The specified DB2 data sharing group member for the group attach name and metadata library are not associated with each other.

System action: None.

User response: Ensure that the DB2 data sharing group member for the group attach name and the metadata library are associated. If the problem persists, contact IBM Software Support.

CCQD362S The *library_name* metadata library is not associated with the specified *group_attach_name* DB2 group attach name.

Explanation: The specified DB2 group attach name and metadata library are not associated with each other.

System action: None.

User response: Ensure that the DB2 group attach name and the metadata library are associated. If the problem persists, contact IBM Software Support.

CCQD400W The customization parser issued the *code_number* warning code while it parsed the product customization member *member_name*. See the PL/I programming guide for more information about this XML parser continuable exception code.

Explanation: While determining if the specified member is valid, the PL/I XML parser issued an exception warning code.

System action: Processing stops.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the warning.

CCQD401S The customization parser issued the *code_number* error code while it parsed the product customization member *member_name*. See the PL/I programming guide for more information about this XML parser terminating exception code.

Explanation: While determining if the specified member is valid, the PL/I XML parser issued an exception error code.

System action: Processing stops.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the error.

CCQD500W The *data_set_name* data store data set was not found.

Explanation: Tools Customizer could not find the specified data store data set.

System action: None.

User response: No action is required.

CCQD501W The *data_set_name* data store data set was not found, so it was created.

Explanation: Tools Customizer created the specified data set because it could not be found.

System action: None.

User response: No action is required.

CCQD502E The *data_set_name* data store data set is not writable.

Explanation: Tools Customizer cannot write to the specified data set.

System action: None.

User response: Ensure that the data set is writable.

CCQD503E The *data_set_name* data store data set could not be opened with the *disposition_type* disposition.

Explanation: Tools Customizer could not open the data set with the specified disposition.

System action: Processing stops.

User response: Ensure that you have WRITE authority access to this data set.

CCQD504E The *data_set_name* data store data set could not be opened with the *option_name* option.

Explanation: Tools Customizer could not open the data set with the specified option.

System action: Processing stops.

User response: Ensure that you have WRITE authority access to this data set.

CCQD505E The *data_set_name* data store data set could not be created.

Explanation: Tools Customizer could not create the specified data set.

System action: Processing stops.

User response: Ensure that you have the authority to create data sets and that the DASD is not full.

CCQD510I The DB2 SSID and DB2 group attach name were created.

Explanation: The DB2 SSID and DB2 group attach name were created and saved in the data store.

System action: None.

User response: No action is required.

CCQD511E The DB2 entry already exists in the list of DB2 entries to be associated.

Explanation: The DB2 entry cannot be added because it already exists in the list of DB2 entries to be associated.

System action: None.

User response: Specify a different DB2 entry.

CCQD512S An error occurred while a DB2 entry was being created.

Explanation: A severe error occurred while a DB2 entry was being created.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD513E The specified DB2 entry already exists and is associated with the current product on the Customizer Workplace panel.

Explanation: The DB2 entry cannot be added because it already exists, and it is already associated with the product to be customized.

System action: None.

User response: Press F3 to go to the Customizer Workplace panel to see the DB2 entry, or specify a different DB2 entry.

CCQD514E A value is required for a DB2 subsystem, a DB2 group attach name, or both before they can be created.

Explanation: Required information is missing. A DB2 subsystem, a DB2 group attach name, or both must be specified.

System action: None.

User response: Specify a DB2 subsystem, a DB2 group attach name, or both.

CCQD515E The specified DB2 entry already exists in the list of DB2 entries and is already associated with the current product.

Explanation: The DB2 entry has already been created and associated with the product that you want to customize.

System action: None.

User response: Specify a different DB2 entry.

CCQD516E The specified DB2 entry already exists in the list of DB2 entries on the Associate DB2 Entry with Product panel but is not associated with the current product.

Explanation: The DB2 entry exists, but it must be associated with the product to be customized.

System action: None.

User response: On the Customizer Workplace panel, issue the ASSOCIATE command to associate the DB2 entry with the product.

CCQD517S An error occurred while a DB2 entry was being copied.

Explanation: A severe error occurred while a DB2 entry was being copied

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD518E A value is required for a DB2 subsystem, a DB2 group attach name, or both before they can be copied.

Explanation: Required information is missing. A DB2 subsystem, a DB2 group attach name, or both must be specified.

System action: None.

User response: Specify a DB2 subsystem, a DB2 group attach name, or both.

CCQD519I The DB2 entry was copied.

Explanation: The DB2 entry was copied and saved in the Tools Customizer data store.

System action: None.

User response: No action is required.

CCQD520S The DB2 entry was copied to the list of DB2 entries but was not associated because the product is already associated with the allowed number of DB2 entries.

Explanation: The DB2 entry was not completely copied because a product can be associated with only 1200 DB2 entries.

System action: Processing stops.

User response: Remove a DB2 entry from the list, and copy the specified DB2 entry again.

CCQD521E *Line_command* is not a valid line command.

Explanation: The specified line command is not valid. Valid line commands are on the panel.

System action: Processing stops.

User response: Specify a valid line command.

CCQD522E The *subsystem_ID* DB2 subsystem ID occurs more than once in the list. Each row must be unique.

Explanation: The specified DB2 subsystem ID can be used only once.

System action: Processing stops.

User response: Specify a different DB2 subsystem ID.

CCQD523E The *group_attach_name* DB2 group attach name occurs more than once in the list. Each row must be unique.

Explanation: The specified DB2 group attach name can be used only once.

System action: Processing stops.

User response: Specify a different DB2 group attach name.

CCQD524E The *member_name* DB2 member for the DB2 group attach name occurs more than once in the list. Each row must be unique.

Explanation: The specified DB2 member for the DB2 group attach name can be used only once.

System action: Processing stops.

User response: Specify a different DB2 member for the DB2 group attach name.

CCQD525I The DB2 entries were created.

User response: No action is required.

CCQD526E The *subsystem_ID* DB2 subsystem ID occurs more than once in the list. Each DB2 subsystem ID must be unique.

Explanation: The specified DB2 subsystem ID can be used only once.

System action: Processing stops.

User response: Specify a different DB2 subsystem ID.

CCQD527I DB2 group attach names cannot be created during the copy process.

Explanation: The ability to create DB2 group attach names is not available during the copy process.

System action: None.

User response: Create DB2 group attach names by issuing the CREATE command on the Customizer Workplace panel.

CCQD528E The *metadata_library* metadata library is already associated with *number* DB2 entries. The maximum number of associated DB2 entries for this metadata library is 256.

Explanation: A metadata library can be associated with a maximum of 256 DB2 entries. The specified metadata library is already associated with 256.

System action: Processing stops.

User response: Remove an existing association between the specified metadata library and a DB2 entry, and associate the specified the metadata library with another entry.

CCQD529I At least one row is required.

CCQD560E The *subsystem_ID* DB2 subsystem already exists and is associated with the current product on the Customizer Workplace panel.

Explanation: The specified DB2 subsystem exists and is associated with the product that you are customizing.

System action: None.

User response: Specify another DB2 subsystem.

CCQD561E The *member_name* DB2 member for the *group_attach_name* DB2 group attach name already exists and is associated with the current product on the Customizer Workplace panel.

Explanation: The specified DB2 data sharing group for the DB2 group attach name exists and is associated with the product that you are customizing.

System action: None.

User response: Specify another DB2 subsystem.

CCQD562E The *group_attach_name* DB2 group attach name already exists and is associated with the current product on the Customizer Workplace panel.

Explanation: The specified DB2 group attach name exists and is associated with the product that you are customizing. The subsystem is in the table on the Customizer Workplace panel.

System action: None.

User response: Specify another DB2 group attach name.

CCQD563E A value is required for a DB2 subsystem, a DB2 group attach name, or both before they can be created.

Explanation: A DB2 subsystem, a DB2 group attach name, or both are not specified so one or both of them cannot be created.

System action: None.

User response: Specify a value for the DB2 subsystem, the DB2 group attach name, or both.

CCQD565E The *subsystem_ID* DB2 subsystem already exists in the list of DB2 entries and is already associated with the current product.

Explanation: The specified subsystem is already associated.

System action: None.

User response: Specify a different DB2 subsystem.

CCQD566E The *member_name* DB2 member for the *group_attach_name* DB2 group attach name already exists in the list of DB2 entries and is already associated with the current product.

Explanation: The specified DB2 member is already associated.

System action: None.

User response: Specify a different DB2 member.

CCQD567E The *group_attach_name* DB2 group attach name already exists in the list of DB2 entries and is already associated with the current product.

Explanation: The specified DB2 group attach name is already associated.

System action: None.

User response: Specify another DB2 group attach name.

CCQD568I To customize *product_name*, at least one DB2 entry must be associated with this product.

Explanation: The specified product requires at least one associated DB2 entry.

System action: None.

User response: To continue the customization process for the specified product, associate one or more DB2 entries with it.

CCQD569I To customize the *product_name* product configuration, at least one DB2 entry must be associated with this configuration.

Explanation: The configuration for the specified product requires at least one associated DB2 entry.

System action: None.

User response: To continue the customization process for the configuration of the specified product, associate one or more DB2 entries with the configuration.

CCQD577W The *mode_name* DB2 mode of the *subsystem_ID* DB2 subsystem is not supported by the product.

Explanation: The product does not support the specified DB2 mode.

System action: None.

User response: Specify a supported DB2 mode.

CCQD578W The *mode_name* DB2 mode of the *member_name* DB2 member for the DB2 group is not supported by the product.

Explanation: The product does not support the specified DB2 mode.

System action: None.

User response: Specify a supported DB2 mode.

CCQD579W The *mode_name* DB2 mode of the *group_name* DB2 group attach name is not supported by the product.

Explanation: The product does not support the specified DB2 mode.

System action: None.

User response: Specify a supported DB2 mode.

CCQD580S The *subsystem_ID* DB2 subsystem was copied to the list of DB2 entries but was not associated because the product is already associated with the allowed number of DB2 entries.

Explanation: The copied DB2 subsystem was not associated with the product because the product is associated with the maximum number of DB2 entries.

System action: None.

User response: Remove an associated DB2 entry and associate the product with the copied DB2 subsystem.

CCQD581S The *member_name* DB2 member for the *group_attach_name* DB2 group attach name was copied to the list of DB2 entries but was not associated because the product is already associated with the allowed number of DB2 entries.

Explanation: The copied DB2 member for the DB2 group attach name was not associated with the product because the product is associated with the maximum number of DB2 entries.

System action: None.

User response: Remove an associated DB2 entry and associate the product with the copied DB2 member.

CCQD582S The *group_attach_name* DB2 group attach name was copied to the list of DB2 entries but was not associated because the product is already associated with the allowed number of DB2 entries.

Explanation: The copied DB2 group attach name was not associated with the product because the product is

associated with the maximum number of DB2 entries.

System action: None.

User response: Remove an associated DB2 entry and associate the product with the copied DB2 group attach name.

CCQD584I The *member_name* DB2 member for the *group_attach_name* DB2 group attach name is copied to the *subsystem_ID* DB2 subsystem.

Explanation: The specified DB2 member was copied.

System action: None.

User response: No action is required.

CCQD585I The *group_attach_name* DB2 group attach name cannot be copied because a DB2 member is required.

Explanation: The specified DB2 group attach name was not copied because a DB2 member was missing.

System action: None.

User response: No action is required.

CCQD586S The current LPAR is *LPAR_name*, but the data store contains information about the *LPAR_name* LPAR. You must use the *LPAR_name* LPAR to customize the product.

Explanation: The LPAR that is stored in the data store data set must be used to customize the product.

System action: Processing stops.

User response: Use the LPAR that is stored in the data store data set.

CCQD587W The *level_number* DB2 level of the *subsystem_name* DB2 subsystem is not supported by the product.

Explanation: The product does not support the specified DB2 level.

System action: Processing continues.

User response: Specify a supported level of DB2.

CCQD588W The *level_number* DB2 level of the *member_name* DB2 member of the *group_name* DB2 group is not supported by the product.

Explanation: The product does not support the specified DB2 level.

System action: Processing continues.

User response: Specify a supported level of DB2.

CCQD589W The *level_number* DB2 level of the *group_name* DB2 group attach name is not supported by the product.

Explanation: The product does not support the specified DB2 level.

System action: Processing continues.

User response: Specify a supported level of DB2.

CCQD593I The *subsystem_ID* DB2 subsystem was deleted.

User response: No action is required.

CCQD594I The *member_name* DB2 for the *group_attach_name* DB2 group attach name was deleted.

User response: No action is required.

CCQD595I The *group_attach_name* DB2 group attach name was deleted.

User response: No action is required.

CCQD596E The *subsystem_ID* DB2 subsystem was not deleted.

Explanation: An internal error occurred while the specified DB2 subsystem was being deleted.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD597E The *member_name* DB2 member for the *group_attach_name* DB2 group attach name was not deleted.

Explanation: An internal error occurred while the specified DB2 member was being deleted.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD598E The *group_attach_name* DB2 group attach name was not deleted.

Explanation: An internal error occurred while the specified DB2 group attach name was being deleted.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD600W The *member_name* product customization member is not valid. The PL/I XML parser issued the following exception warning code: *code_number*.

Explanation: While determining if the XML structure of the product customization member is valid, the PL/I XML parser issued an exception warning code.

System action: Processing continues.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

CCQD601S The *member_name* product customization member is not valid. The PL/I XML parser issued the following exception error code: *code_number*.

Explanation: While determining if the XML structure of the product customization member is valid, the PL/I XML parser issued an exception error code.

System action: Processing stops.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception error code.

CCQD602S The XML structure of the *member_name* product customization member is not valid. The *element_name* element is unknown.

Explanation: The data store member contains an unknown element.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD603S The XML structure of the *member_name* product customization member is not valid. Content is not allowed for the *element_name* element, but content was found.

Explanation: The specified element cannot contain content.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD604S The XML structure of the *member_name* product customization member is not valid. Content is required for the *element_name* element, but content was not found.

Explanation: The specified element is missing required content.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD605S The XML structure of the *member_name* product customization member is not valid. The content length for the *element_name* element exceeds *maximum_number* characters.

Explanation: The specified element contains too many characters.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD606S The XML structure of the *member_name* product customization member is not valid. The *element_name* element cannot occur more than *maximum_number* times.

Explanation: The specified element occurs too many times.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD607S The XML structure of the *member_name* product customization member is not valid. The *element_name* element must occur at least *minimum_number* times.

Explanation: The specified element does not occur enough times.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD608S The XML structure of the *member_name* product customization member is not valid. The *attribute_name* attribute in the *element_name* element cannot occur more than *maximum_number* times.

Explanation: The specified attribute occurs too many times.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD609S The XML structure of the *member_name* product customization member is not valid. The *attribute_name* attribute in the *element_name* element must occur at least *minimum_number* times.

Explanation: The specified attribute does not occur enough times.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD610S The XML structure of the *member_name* product customization member is not valid. Content is not allowed for the *attribute_name* attribute in the *element_name* element, but content was found.

Explanation: The specified attribute cannot contain content.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD611S The XML structure of the *member_name* product customization member is not valid. Content is required for the *attribute_name* attribute in the *element_name* element, but content was not found.

Explanation: The specified attribute does not contain required content.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD612S The XML structure of the *member_name* product customization member is not valid. The content length for the *element_name* element exceeds *maximum_number* characters.

Explanation: The specified element contains too many characters.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD613S The XML structure of the *member_name* product customization member is not valid. The *attribute_name* attribute in the *element_name* element is unknown.

Explanation: The specified attribute in the data store member is unknown.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD614S The content of the *member_name* product customization member is not valid. The value of the *element_name* element is not valid. The value is *value_name*.

Explanation: The specified value is not valid.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQD700W The *member_name* DB2 data member is not valid. The PL/I XML parser issued the following exception warning code: *code_number*.

Explanation: While determining if the XML structure of the DB2 data member is valid, the PL/I XML parser issued an exception warning code.

System action: Processing continues.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

CCQD701S The *member_name* DB2 data member is not valid. The PL/I XML parser issued the following exception error code: *code_number*.

Explanation: While determining if the XML structure of the DB2 data member is valid, the PL/I XML parser issued an exception error code.

System action: Processing continues.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception error code.

CCQD750W The *value_number* value in the DB2 parameter *parameter_name* was skipped because only *maximum_number* values are allowed.

Explanation: The specified value was skipped because it exceeds the number of allowed values in the DB2 parameter.

System action: Processing continues.

User response: No action is required. To stop this message from being issued, remove the extra values from the DB2 parameter.

CCQD800W The *member_name* LPAR data member is not valid. The PL/I XML parser issued the following exception warning code: *code_number*.

Explanation: While determining if the XML structure of the LPAR data member is valid, the PL/I XML parser issued an exception warning code.

System action: Processing continues.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

CCQD801S The *member_name* LPAR data member is not valid. The PL/I XML parser issued the following exception error code: *code_number*.

Explanation: While determining if the XML structure of the LPAR data member is valid, the PL/I XML parser issued an exception error code.

System action: Processing continues.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception error code.

CCQD850W The *value_number* value in the LPAR parameter *parameter_name* was skipped because only *maximum_number* values are allowed.

Explanation: The specified value was skipped because it exceeds the number of allowed values in the LPAR parameter.

System action: Processing continues.

User response: No action is required. To stop this message from being issued, remove the extra values from the LPAR parameter.

CCQD851I The *subsystem_ID* DB2 subsystem is copied to the *member_name* DB2 member for the *group_attach_name* DB2 group attach name.

User response: No action is required.

CCQD852I The *member_name* DB2 member for the *group_attach_name* DB2 group attach name is copied to the *member_name* DB2 member for the *group_attach_name* DB2 group attach name.

User response: No action is required.

CCQD854I The *member_name* DB2 member for the *group_attach_name* DB2 group 'attach name is copied to multiple DB2 entries.

User response: No action is required.

CCQD900W The *member_name* product data member is not valid. The PL/I XML parser issued the following exception warning code: *code_number*.

Explanation: While determining if the XML structure of the product data member is valid, the PL/I XML parser issued an exception warning code.

System action: Processing continues.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

CCQD901S The *member_name* product data member is not valid. The PL/I XML parser issued the following exception error code: *code_number*.

Explanation: While determining if the XML structure of the product data member is valid, the PL/I XML parser issued an exception error code.

System action: Processing continues.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

CCQD950W The *value_number* value in the product parameter *parameter_name* was skipped because only *maximum_number* values are allowed.

Explanation: The specified value was skipped because it exceeds the number of allowed values in the product parameter.

System action: Processing continues.

User response: No action is required. To stop this message from being issued, remove the extra values from the product parameter.

CCQD960I The *subsystem_ID* DB2 subsystem was changed to the *member_name* DB2 member for the *group_attach_name* DB2 group attach name.

User response: No action is required.

CCQD961I The *member_name* DB2 member for the *group_attach_name* DB2 group attach name was changed to the *subsystem_ID* DB2 subsystem.

User response: No action is required.

CCQD962I The *member_name* DB2 member for the *group_attach_name* DB2 group attach name was changed to the *member_name* DB2 member for the *group_attach_name* DB2 group attach name.

User response: No action is required.

CCQD963E The DB2 group attach name cannot be blank when the DB2 subsystem ID is blank.

Explanation: A DB2 group attach name, DB2 subsystem ID, or both must be specified.

System action: Processing stops.

User response: Specify a DB2 group attach name, DB2 subsystem ID, or both.

CCQE000S The specified message field name or message *message_ID* was not found.

Explanation: An error occurred while displaying a message field name or the specified message.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQE001E An incorrect trace level was specified. Valid trace levels are 0 - 4.

Explanation: A wrong trace level was specified. Valid trace levels are 0 - 4.

System action: Processing stops.

User response: Specify a valid trace level 0 - 4.

CCQH001W The specified option *option_name* is not valid.

Explanation: The option that was specified is not a valid option on the panel.

System action: Tools Customizer stops.

User response: Specify a valid option on the panel.

CCQH006W Before you customize a product, verify your user settings.

Explanation: The user settings must be verified before a product can be customized.

System action: Tools Customizer stops.

User response: Verify the user settings.

CCQH007E Check the user settings. One or more current values are not valid.

Explanation: One or more of the values in the user settings is not valid.

System action: Tools Customizer stops.

User response: Ensure that the specified values for the user settings are valid.

CCQH008W Before you use Tools Customizer, you must select option 0 to verify your user settings.

Explanation: The user settings must be changed before a product can be customized.

System action: Tools Customizer stops.

User response: Change the user settings.

CCQH009E You must select option 0 to change your user settings.

Explanation: User settings must be changed before a product can be customized.

System action: Tools Customizer stops.

User response: Change the user settings.

CCQI000W The XML structure of the *member_name* DB2 parameter metadata member is not valid. The PL/I XML parser issued the following exception warning code: *code_number*.

Explanation: While determining if the DB2 parameter metadata member is valid, the PL/I XML parser issued an exception warning code.

System action: Processing continues.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

CCQI001S The XML structure of the *member_name* DB2 parameter metadata member is not valid. The PL/I XML parser issued the following exception error code: *code_number*.

Explanation: While determining if the DB2 parameter metadata member is valid, the PL/I XML parser issued an exception error code.

System action: Processing stops.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

CCQI002S The XML structure of the *member_name* DB2 parameter metadata member is not valid. The *element_name* element is unknown.

Explanation: The specified element in the DB2 parameter metadata member is unknown.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI003S The XML structure of the *member_name* DB2 parameter metadata member is not valid. Content is not allowed for the *element_name* element, but content was found.

Explanation: The specified element cannot contain content.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI004S The XML structure of the *member_name* DB2 parameter metadata member is not valid. Content is required for the *element_name* element, but content was not found.

Explanation: The specified element requires content.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI005S The XML structure of the *member_name* DB2 parameter metadata member is not valid. The content length for the *element_name* element cannot exceed *maximum_number* characters.

Explanation: The specified element contains too many characters.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI006S The XML structure of the *member_name* DB2 parameter metadata member is not valid. The content length for the *element_name* element must be at least *minimum_number* characters.

Explanation: The specified element does not contain enough characters.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI007S The XML structure of the *member_name* DB2 parameter metadata member is not valid. The *element_name* element must occur at least *minimum_number* times.

Explanation: The specified element does not occur enough times.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI008S The XML structure of the *member_name* DB2 parameter metadata member is not valid. The *attribute_name* attribute in the *element_name* element cannot occur more than *maximum_number* times.

Explanation: The specified attribute occurs too many times.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI009S The XML structure of the *member_name* DB2 parameter metadata member is not valid. The *attribute_name* attribute in the *element_name* element must occur at least *minimum_number* times.

Explanation: The specified attribute did not occur enough times.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI010S The XML structure of the *member_name* DB2 parameter metadata member is not valid. Content is not allowed for the *attribute_name* attribute in the *element_name* element, but content was found.

Explanation: The specified attribute cannot have content.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI011S The XML structure of the *member_name* DB2 parameter metadata member is not valid. Content is required for the *attribute_name* attribute in the *element_name* element, but content was not found.

Explanation: The specified attribute is missing required content.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI012S The XML structure of the *member_name* DB2 parameter metadata member is not valid. The content length for the *element_name* element cannot exceed *maximum_number* characters.

Explanation: The specified element contains too many characters.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI013S The XML structure of the *member_name* DB2 parameter metadata member is not valid. The *attribute_name* attribute in the *element_name* element is unknown.

Explanation: The specified attribute in the DB2 parameter metadata member is unknown.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI014S The content of the *member_name* DB2 parameter metadata member is not valid because the value of the *element_name* element is incorrect. The value is *value_name*.

Explanation: The specified value of the element is not a valid value.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI015S The content of the DB2 parameter metadata member is not valid because the value of the *attribute_name* attribute in the *element_name* element is incorrect. The value of the attribute is *value_name*.

Explanation: The specified value of the attribute is not a valid value.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI016S The content of the DB2 parameter metadata member is not valid because the data type of the *element_name* element is incorrect. The value is *value_name*.

Explanation: The specified data type is not a valid data type.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI017S The content of the DB2 parameter metadata member is not valid because the data type of the *attribute_name* attribute in the *element_name* element is incorrect. The value of the attribute is *value_name*.

Explanation: The specified data type is not a valid data type.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI050S The *member_name* DB2 parameter metadata member was not found in the *data_set_name* data set.

Explanation: Tools Customizer could not find the specified DB2 parameter metadata member.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI051S The *parameter_name* LPAR parameter in the *template_name* template does not have associated metadata in the *member_name* LPAR parameter metadata member.

Explanation: The specified template does not contain metadata for an LPAR parameter. The name of the LPAR parameter metadata member, the name of the LPAR parameter, and the name of the template are indicated in the message text.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI052S The *parameter_name* product parameter in the *template_name* template does not have associated metadata in the *member_name* product parameter metadata member.

Explanation: The specified template does not contain metadata for a product parameter. The name of the product parameter metadata member, the name of the product parameter, and the name of the template are indicated in the message text.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI053E The following metadata data set was not found: *data_set_name*.

Explanation: Tools Customizer could not find the specified metadata data set.

System action: Processing stops.

User response: Ensure that the metadata data set is specified correctly. If the problem persists, contact IBM Software Support.

CCQI054E The following metadata data set could not be opened: *data_set_name*.

Explanation: Tools Customizer could not open the specified LPAR metadata data set.

System action: Processing stops.

User response: Ensure the metadata data set was specified correctly.

CCQI055S The CCQ\$\$DB2 DB2 parameter metadata member was not found in the *data_set_name* Tools Customizer metadata data set.

Explanation: Tools Customizer could not find the DB2 parameter metadata member in the specified Tools Customizer metadata data set.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI056S The CCQ\$\$LPR LPAR parameter metadata member was not found in the *data_set_name* data set.

Explanation: Tools Customizer could not find the specified LPAR parameter metadata member.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI057S The *member_name* product parameter metadata member was not found in the *data_set_name* data set.

Explanation: The product parameter metadata member was not found in the specified data set.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI058I *Product_name* does not have any DB2 parameters.

Explanation: DB2 parameters are not required to customize the specified product.

System action: Processing continues.

User response: No action is required.

CCQI059I *Product_name* does not have any LPAR parameters.

Explanation: LPAR parameters are not required to customize the specified product.

System action: Processing continues.

User response: No action is required.

CCQI060S The *parameter_name* DB2 parameter in the *task_description* task condition does not have associated metadata in the *member_name* DB2 parameter metadata member.

Explanation: Associated metadata is missing for the specified DB2 parameter in a task.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI061S The *parameter_name* LPAR parameter in the *task_description* task condition does not have associated metadata in the *member_name* LPAR parameter metadata member.

Explanation: Associated metadata is missing for the specified LPAR parameter in a task.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI062S The *parameter_name* product parameter in the *task_description* task condition does not have associated metadata in the *member_name* product parameter metadata member.

Explanation: Associated metadata is missing for the specified product parameter in a task.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI063S The *parameter_name* DB2 parameter in the *task_description* task and the *step_description* step does not have associated metadata in the *member_name* DB2 parameter metadata member.

Explanation: Associated metadata is missing for the specified DB2 parameter in a task and step.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI064S The *parameter_name* LPAR parameter in the *task_description* task and the *step_description* step does not have associated metadata in the *member_name* LPAR parameter metadata member.

Explanation: Associated metadata is missing for the specified LPAR parameter in a task and step.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI065S The *parameter_name* product parameter in the *task_description* task and the *step_description* step does not have associated metadata in the *member_name* parameter metadata member.

Explanation: Associated metadata is missing for the specified parameter in a task and step.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI066S The *parameter_name* DB2 parameter in the *task_description* task, *step_description* step, and *template_name* template condition does not have associated metadata in the *member_name* DB2 parameter metadata member.

Explanation: Associated metadata is missing for the

specified DB2 parameter in a task, step, and template.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI067S The *parameter_name* LPAR parameter in the *task_description* task, *step_description* step, and *template_name* template condition does not have associated metadata in the *member_name* LPAR parameter metadata member.

Explanation: Associated metadata is missing for the specified LPAR parameter in a task, step, and template.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI068S The *parameter_name* product parameter in the *task_description* task, *step_description* step, and *template_name* template condition does not have associated metadata in the *member_name* product parameter metadata member.

Explanation: Associated metadata is missing for the specified product parameter in a task, step, and template.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI069S Product metadata does not support multiple configurations, but the *template_name* product template contains the *parameter_name* parameter. Enable multiple configurations support for this product, and try again.

Explanation: The specified template contains a parameter for multiple configurations, but the product is not enabled to support multiple configurations.

System action: Processing stops.

User response: Enable multiple configurations support, and try again.

CCQI070E The *parameter_name* DB2 parameter metadata member is not valid. The default length for the *parameter-element_name* parameter element exceeds the length of the parameter. The default length is *default_length*, and the specified length is *specified_length*. The default length will be truncated accordingly.

Explanation: The specified length cannot be shorter than the default length.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI071E The *parameter_name* LPAR parameter metadata member is not valid. The default length for the *parameter-element_name* parameter element exceeds the length of the parameter. The default length is *default_length*, and the specified length is *specified_length*. The default length will be truncated accordingly.

Explanation: The specified length cannot be shorter than the default length.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI072E The *parameter_name* product parameter metadata member is not valid. The default length for the *parameter-element_name* parameter element exceeds the length of the parameter. The default length is *default_length*, and the specified length is *specified_length*. The default length will be truncated accordingly.

Explanation: The specified length cannot be shorter than the default length.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI073S The XML structure of the *member_name* DB2 parameter metadata member is not valid. The following value of the *attribute_name* attribute in the *element_name* element already exists: *value_name*.

Explanation: The specified value already exists for an attribute.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI074S The XML structure of the *member_name* LPAR parameter metadata member is not valid. The following value of the *attribute_name* attribute in the *element_name* element already exists: *value_name*.

Explanation: The specified value already exists for an attribute.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI075S The XML structure of the *member_name* product parameter metadata member is not valid. The following value of the *attribute_name* attribute in the *element_name* element already exists: *value_name*.

Explanation: The specified value already exists for an attribute.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI076S The XML structure of the *member_name* DB2 parameter metadata member is not valid. The *parameter_name* parameter refers to the *section-name* section. This section was not found in the DB2 parameter metadata member.

Explanation: The specified value already exists for an attribute.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI077S The XML structure of the *member_name* LPAR parameter metadata member is not valid. The *parameter_name* parameter refers to the *section-name* section. This section was not found in the LPAR parameter metadata member.

Explanation: The specified parameter refers to a section that is not in the LPAR parameter metadata member.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI078S The XML structure of the *member_name* product parameter metadata member is not valid. The *parameter_name* parameter refers to the *section-name* section. This section was not found in the product parameter metadata member.

Explanation: The specified parameter refers to a section that is not in the product parameter metadata member.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI080S The content of the *member_name* DB2 parameter metadata member is not valid because the value of the *attribute_name* attribute in the *element_name* element is incorrect. The value of the attribute is *value_name*.

Explanation: The specified value for an attribute in the DB2 parameter metadata member is not valid.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI081S The content of the *member_name* LPAR parameter metadata member is not valid because the value of the *attribute_name* attribute in the *element_name* element is incorrect. The value of the attribute is *value_name*.

Explanation: The specified value for an attribute in the LPAR parameter metadata member is not valid.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI082S The content of the *member_name* product parameter metadata member is not valid because the value of the *attribute_name* attribute in the *element_name* element is incorrect. The value of the attribute is *value_name*.

Explanation: The specified value for an attribute in the product parameter metadata member is not valid.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI090S The product-defined DB2 parameter *parameter_name* in the *member_name* parameter metadata member references the *section_ID* section ID, but this ID does not exist in either the parameter metadata member or the DB2 parameter metadata member.

Explanation: A section that does not exist in the parameter metadata member or the DB2 parameter metadata member is referenced by the specified DB2 parameter.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI091S The product-defined LPAR parameter in the *member_name* parameter metadata member references the *section_ID* section ID, but this ID does not exist in either the parameter metadata member or the LPAR parameter metadata member.

Explanation: A section that does not exist in the parameter metadata member or the LPAR parameter metadata member is being referenced by the specified LPAR parameter.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI092S The overridden DB2 parameter *parameter_name* in the *member_name* parameter metadata member does not exist in the DB2 parameter metadata member.

Explanation: The specified parameter does not exist.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI093S The overridden LPAR parameter *parameter_name* in the *member_name* parameter metadata member does not exist in the LPAR parameter metadata member.

Explanation: The specified parameter does not exist.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI094S The CCQ\$\$PRD product customization parameter metadata member was not found in the *data_set_name* data set.

Explanation: The specified data set must contain the CCQ\$\$PRD product customization parameter metadata member

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI100W The XML structure of the *member_name* LPAR parameter metadata member is not valid. The PL/I XML parser issued the following exception warning code: *code_number*.

Explanation: While determining if the LPAR parameter metadata member is valid, the PL/I XML parser issued an exception warning code.

System action: Processing continues.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

CCQI101S The XML structure of the *member_name* LPAR parameter metadata member is not valid. The PL/I XML parser issued the following exception error code: *code_number*.

Explanation: While determining if the LPAR parameter metadata member is valid, the PL/I XML parser issued an exception error code.

System action: Processing stops.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

CCQI102S The XML structure of the *member_name* LPAR parameter metadata member is not valid. The *element_name* element is unknown.

Explanation: The specified element in the LPAR parameter metadata member is unknown.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI103S The XML structure of the *member_name* LPAR parameter metadata member is not valid. Content is not allowed for the *element_name* element, but content was found.

Explanation: The specified element cannot contain content.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI104S The XML structure of the *member_name* LPAR parameter metadata member is not valid. Content is required for the *element_name* element, but content was not found.

Explanation: The specified element requires content.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI105S The XML structure of the *member_name* LPAR parameter metadata member is not valid. The content length for the *element_name* element cannot exceed *maximum_number* characters.

Explanation: The specified element contains too many characters.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI106S The XML structure of the *member_name* LPAR parameter metadata member is not valid. The content length for the *element_name* element must be at least *minimum_number* characters.

Explanation: The specified element does not contain enough characters.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI107S The XML structure of the *member_name* LPAR parameter metadata member is not valid. The *element_name* element must occur at least *minimum_number* times.

Explanation: The specified element does not occur enough times.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI108S The XML structure of the *member_name* LPAR parameter metadata member is not valid. The *attribute_name* attribute in the *element_name* element cannot occur more than *maximum_number* times.

Explanation: The specified attribute occurs too many times.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI109S The XML structure of the *member_name* LPAR parameter metadata member is not valid. The *attribute_name* attribute in the *element_name* element must occur at least *minimum_number* times.

Explanation: The specified attribute did not occur enough times.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI110S The XML structure of the *member_name* LPAR parameter metadata member is not valid. Content is not allowed for the *attribute_name* attribute in the *element_name* element, but content was found.

Explanation: The specified attribute cannot have content.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI111S The XML structure of the *member_name* LPAR parameter metadata member is not valid. Content is required for the *attribute_name* attribute in the *element_name* element, but content was not found.

Explanation: The specified attribute is missing required content.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI112S The XML structure of the *member_name* LPAR parameter metadata member is not valid. The content length for the *element_name* element cannot exceed *maximum_number* characters.

Explanation: The specified element contains too many characters.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI113S The XML structure of the *member_name* LPAR parameter metadata member is not valid. The *attribute_name* attribute in the *element_name* element is unknown.

Explanation: The specified attribute in the LPAR parameter metadata member is unknown.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI114S The content of the *member_name* LPAR parameter metadata member is not valid because the value of the *element_name* element is incorrect. The value is *value_name*.

Explanation: The specified value for an element in the LPAR parameter metadata member is not valid.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI115S The content of the *member_name* LPAR parameter metadata member is not valid because the value of the *attribute_name* attribute in the *element_name* element is incorrect. The value of the attribute is *value_name*.

Explanation: The specified value for an attribute in the LPAR parameter metadata member is not valid.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI116S The content of the *member_name* LPAR parameter metadata member is not valid because the data type of the *element_name* element is incorrect. The value is *value_name*.

Explanation: The specified data type value for an element in the LPAR parameter metadata member is not valid.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI117S The content of the *member_name* LPAR parameter metadata member is not valid because the data type of the *attribute_name* attribute in the *element_name* element is incorrect. The value is *value_name*.

Explanation: The specified data type value for an attribute in the LPAR parameter metadata member is not valid.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI120S The XML structure of the *member_name* DB2 parameter metadata member is not valid. The *element_name* element in the *parameter_name* parameter contains duplicate values for the *element_name* element. The duplicate value is *value_name*.

Explanation: An element contains the specified duplicate value.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI121S The XML structure of the *member_name* LPAR parameter metadata member is not valid. The *element_name* element in the *parameter_name* parameter contains duplicate values for the *element_name* element. The duplicate value is *value_name*.

Explanation: An element contains the specified duplicate value.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI122S The XML structure of the *member_name* parameter metadata member is not valid. The *element_name* element in the *parameter_name* parameter contains duplicate values for the *element_name* element. The duplicate value is *value_name*.

Explanation: An element contains the specified duplicate value.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI123S The XML structure of the *member_name* discover metadata member is not valid. The *element_name* element in the *parameter_name* parameter contains duplicate values for the *element_name* element. The duplicate value is *value_name*.

Explanation: An element contains the specified duplicate value.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI124S The XML structure of the *member_name* product customization parameter metadata member is not valid. The *element_name* element in the *parameter_name* parameter contains duplicate values for the *element_name* element. The duplicate value is *value_name*.

Explanation: An element contains the specified duplicate value.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI200W The XML structure of the *member_name* information metadata member is not valid. The PL/I XML parser issued the following exception warning code: *code_number*.

Explanation: While determining if the information metadata member is valid, the PL/I XML parser issued an exception warning code.

System action: Processing continues.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

CCQI201S The XML structure of the *member_name* information metadata member is not valid. The PL/I XML parser issued the following exception error code: *code_number*.

Explanation: While determining if the information metadata member is valid, the PL/I XML parser issued an exception error code.

System action: Processing stops.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

CCQI202S The XML structure of the *member_name* information metadata member is not valid. The *element name* element is unknown.

Explanation: The specified element in the information metadata member is unknown.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI203S The XML structure of the *member_name* information metadata member is not valid. Content is not allowed for the *element_name* element, but content was found.

Explanation: The specified element cannot contain content.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI204S The XML structure of the *member_name* information metadata member is not valid. Content is required for the *element_name* element, but content was not found.

Explanation: The specified element requires content.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI205S The XML structure of the *member_name* information metadata member is not valid. The content length for the *element_name* element cannot exceed *maximum_number* characters.

Explanation: The specified element contains too many characters.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI206S The XML structure of the *member_name* information metadata member is not valid. The content length for the *element_name* element must be at least *minimum_number* characters.

Explanation: The specified element does not contain enough characters.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI207S The XML structure of the *member_name* information metadata member is not valid. The *element_name* element must occur at least *minimum_number* times.

Explanation: The specified element does not occur enough times.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI208S The XML structure of the *member_name* information metadata member is not valid. The *attribute_name* attribute in the *element_name* element cannot occur more than *maximum_number* times.

Explanation: The specified attribute occurs too many times.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI209S The XML structure of the *member_name* information metadata member is not valid. The *attribute_name* attribute in the *element_name* element must occur at least *minimum_number* times.

Explanation: The specified attribute did not occur enough times.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI210S The XML structure of the *member_name* information metadata member is not valid. Content is not allowed for the *attribute_name* attribute in the *element_name* element, but content was found.

Explanation: The specified attribute cannot have content.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI211S The XML structure of the *member_name* information metadata member is not valid. Content is required for the *attribute_name* attribute in the *element_name* element, but content was not found.

Explanation: The specified attribute is missing required content.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI212S The XML structure of the *member_name* information metadata member is not valid. The content length for the *element_name* element cannot exceed *maximum_number* characters.

Explanation: The specified element contains too many characters.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI213S The XML structure of the *member_name* information metadata member is not valid. The *attribute_name* attribute in the *element_name* element is unknown.

Explanation: The specified attribute in the information metadata member is unknown.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI214S The content of the *member_name* information metadata member is not valid because the value of the *element_name* element is incorrect. The value is *value_name*.

Explanation: The specified value for an element in the information metadata member is not valid.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI215S The content of the *member_name* information metadata member is not valid because the value of the *attribute_name* attribute in the *element_name* element is incorrect. The value is *value_name*.

Explanation: The specified value for an attribute in

the information metadata member is not valid.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI216S The content of the *member_name* information metadata member is not valid because the data type of the *element_name* element is incorrect. The value is *value_name*.

Explanation: The specified data type value for an element in the information metadata member is not valid.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI217S The content of the *member_name* information metadata member is not valid because the data type of the *attribute_name* attribute in the *element_name* element is incorrect. The value is *value_name*.

Explanation: The specified data type value for an attribute in the information metadata member is not valid.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI218S The content of the *member_name* information metadata member is not valid. The length of the *value_name* value that of the *attribute_name* attribute is longer than the *value_name* value of the *attribute_name* attribute.

Explanation: The first specified value cannot be longer than the second specified value.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI219S The content of the *member_name* information metadata member is not valid. The *value_name* value of the *attribute_name* attribute contains the *value_name* value.

Explanation: The first specified value cannot be longer than the second specified value.

System action: Processing stops.

User response: See Gathering diagnostic information.

Contact IBM Software Support.

CCQI220S The XML structure of the *member_name* information metadata member is not valid. Content for the *attribute_name* attribute in the *element_name* element exceed *maximum_number* characters.

Explanation: The specified attribute contains too many characters.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI223S The XML structure of the *member_name* information metadata member is not valid. The value that is specified for the DB2 Level already exists. The value is *value_name*.

Explanation: The specified value already exists.

System action: Processing stops.

User response: Specify a different DB2 level. If the problem persists, contact IBM Software Support.

CCQI224S The XML structure of the *member_name* information metadata member is not valid. The value that is specified for the DB2 Mode already exists. The value is *value_name*.

Explanation: The specified value already exists.

System action: Processing stops.

User response: Specify a different DB2 mode. If the problem persists, contact IBM Software Support.

CCQI250S The information metadata member was not found in the *data_set_name* data set.

Explanation: Tools Customizer could not find the information metadata member in the specified data set.

System action: Processing stops.

User response: If this message was issued on the Specify the Metadata Library (CCQPHLQ) panel, specify the product metadata library. The name of this library is *hlq.SCKZDENU*.

Do not specify the Tools Customizer metadata library, which is *hlq.SCCQDENU*.

If the problem persists, identify the name of the Tools Customizer trace data set and contact IBM Software Support.

CCQI251E The *member_name* member was not accessible in the *data_set_name* data set.

Explanation: The specified member could not be accessed in the data set.

System action: Processing stops.

User response: Specify the correct metadata library.

CCQI252S The information metadata member was not found in the *library_name* component metadata library that is part of the *library_name* pack metadata library. The name of the pack is *pack_name*.

Explanation: The specified component metadata library does not contain the information metadata member.

System action: Processing stops.

User response: Specify the correct metadata library.

CCQI253E The *library_name* Tools Customizer metadata library is not current. Update the metadata library on the Tools Customizer Settings panel.

Explanation: The specified metadata library is not current.

System action: Processing stops.

User response: Specify a current metadata library on the Tools Customizer Settings panel.

CCQI300W The XML structure of the *member_name* sequence metadata member is not valid. The PL/I XML parser issued the following exception warning code: *code_number*.

Explanation: While determining if the sequence metadata member is valid, the PL/I XML parser issued an exception warning code.

System action: Processing continues.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

CCQI301S The XML structure of the *member_name* sequence metadata member is not valid. The PL/I XML parser issued the following exception error code: *code_number*.

Explanation: While determining if the sequence metadata member is valid, the PL/I XML parser issued an exception error code.

System action: Processing stops.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception error code, and contact IBM Software Support.

CCQI302S The XML structure of the *member_name* sequence metadata member is not valid. The *element_name* element is unknown.

Explanation: The specified element in the sequence metadata member is unknown.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI303S The XML structure of the *member_name* sequence metadata member is not valid. Content is not allowed for the *element_name* element, but content was found.

Explanation: The specified element cannot contain content.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI304S The XML structure of the *member_name* sequence metadata member is not valid. Content is required for the *element_name* element, but content was not found.

Explanation: The specified element is missing required content.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI305S The XML structure of the *member_name* sequence metadata member is not valid. Content length for the *element_name* element cannot exceed *maximum_number* characters.

Explanation: The specified element contains too many characters.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI306S The XML structure of the *member_name* sequence metadata member is not valid. The *element_name* element cannot occur more than *maximum_number* times.

Explanation: The specified element occurs too many times.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI307S The XML structure of the *member_name* sequence metadata member is not valid. The *element_name* element must occur at least *minimum_number* times.

Explanation: The specified element does not occur enough times.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI308S The XML structure of the *member_name* sequence metadata member is not valid. The *attribute_name* attribute in the *element_name* element cannot occur more than *maximum_number* times.

Explanation: The specified attribute occurs too many times.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI309S The XML structure of the *member_name* sequence metadata member is not valid. The *attribute_name* attribute in the *element_name* element must occur at least *minimum_number* times.

Explanation: The specified attribute does not occur enough times.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI310S The XML structure of the *member_name* sequence metadata member is not valid. Content is not allowed for the *attribute_name* attribute in the *element_name* element, but content was found.

Explanation: The specified attribute cannot contain content.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI311S The XML structure of the *member_name* sequence metadata member is not valid. Content is required for the *attribute_name* attribute in the *element_name* element, but content was not found.

Explanation: The specified attribute is missing required content.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI312S The XML structure of the *member_name* sequence metadata member is not valid. The content length for the *element_name* element cannot exceed *maximum_number* characters.

Explanation: The specified element contains too many characters.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI313S The XML structure of the *member_name* sequence metadata member is not valid. The *attribute_name* attribute in the *element_name* element is unknown.

Explanation: The specified attribute in the sequence metadata member is unknown.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI314S The content of the *member_name* sequence metadata member is not valid because the value of the *element_name* element is incorrect. The value is *value_name*.

Explanation: The specified value for an element in the sequence metadata member is not valid.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI315S The content of the *member_name* sequence metadata member is not valid because the value of the *attribute_name* attribute in the *element_name* element is incorrect. The value is *value_name*.

Explanation: The specified value for an attribute in the sequence metadata member is not valid.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI316S The content of the *member_name* sequence metadata member is not valid because the data type of the *element_name* element is incorrect. The value is *value_name*.

Explanation: The specified data type value for an element in the sequence metadata member is not valid.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI317S The content of the *member_name* sequence metadata member is not valid because the data type of the *attribute_name* attribute in the *element_name* element is incorrect. The value is *value_name*.

Explanation: The specified data type value for an attribute in the sequence metadata member is not valid.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI350S The XML structure of the *member_name* sequence metadata member is not valid because the value of the *attribute_name* attribute in the *element_name* element is incorrect. The value is *value_name*.

Explanation: A specified value for an attribute in the sequence metadata member is not valid.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI351S The *member_name* sequence metadata member was not found in the *data_set_name* metadata data set.

Explanation: Tools Customizer could not find the specified sequence metadata member in the metadata data set.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI352S The *template_name* product template was not found in the *data_set_name* metadata data set.

Explanation: Tools Customizer could not find the specified product template in the data set.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI353S The sequence metadata member was not found in the *data_set_name* component data set that is part of the *data_set_name* pack.

Explanation: Tools Customizer could not find the sequence metadata member.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI360S The XML structure of the *member_name* sequence metadata member is not valid. The value of the *attribute_name* attribute in the *element_name* element already exists.

Explanation: The specified attribute contains a value that already exists.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI361S The XML structure of the *member_name* sequence metadata member is not valid. The condition element on the *level_type* level already contains a relational operator.

Explanation: A relational operator already exists for the condition element on the specified level.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI362S The XML structure of the *member_name* sequence metadata member is not valid. The condition element on the *level_type* level must contain only one content string or content number element.

Explanation: Only one content string element or content number element can be contained in the condition element on the specified level.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI363S The XML structure of the *member_name* sequence metadata member is not valid. The condition element in the *element_name* element with the *attribute_name* attribute must contain either the content string element or content number element.

Explanation: Either the content string element or the content number element must be in the condition element.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI400W The XML structure of the *member_name* parameter metadata member is not valid. The PL/I XML parser issued the following exception warning code: *code_number*.

Explanation: While determining the parameter metadata member is valid, the PL/I XML parser issued an exception warning code.

System action: Processing continues.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

CCQI401S The XML structure of the *member_name* parameter metadata member is not valid. The PL/I XML parser issued the following exception error code: *code_number*.

Explanation: While determining if the parameter metadata member is valid, the PL/I XML parser issued an exception error code.

System action: Processing stops.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

CCQI402S The XML structure of the *member_name* parameter metadata member is not valid. The *element name* element is unknown.

Explanation: The specified element in the parameter metadata member is unknown.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI403S The XML structure of the *member_name* parameter metadata member is not valid. Content is not allowed for the *element_name* element, but content was found.

Explanation: The specified element cannot contain content.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI404S The XML structure of the *member_name* parameter metadata member is not valid. Content is required for the *element_name* element, but content was not found.

Explanation: The specified element requires content.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI405S The XML structure of the *member_name* parameter metadata member is not valid. The content length for the *element_name* element cannot exceed *maximum_number* characters.

Explanation: The specified element contains too many characters.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI406S The XML structure of the *member_name* parameter metadata member is not valid. The content length for the *element_name* element must be at least *minimum_number* characters.

Explanation: The specified element does not contain enough characters.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI407S The XML structure of the *member_name* parameter metadata member is not valid. The *element_name* element must occur at least *minimum_number* times.

Explanation: The specified element does not occur enough times.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI408S The XML structure of the *member_name* parameter metadata member is not valid. The *attribute_name* attribute in the *element_name* element cannot occur more than *maximum_number* times.

Explanation: The specified attribute occurs too many times.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI409S The XML structure of the *member_name* parameter metadata member is not valid. The *attribute_name* attribute in the *element_name* element must occur at least *minimum_number* times.

Explanation: The specified attribute does not occur enough times.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI410S The XML structure of the *member_name* parameter metadata member is not valid. Content is not allowed for the *attribute_name* attribute in the *element_name* element, but content was found.

Explanation: The specified attribute cannot have content.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI411S The XML structure of the *member_name* parameter metadata member is not valid. Content is required for the *attribute_name* attribute in the *element_name* element, but content was not found.

Explanation: The specified attribute is missing required content.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI412S The XML structure of the *member_name* parameter metadata member is not valid. The content length for the *element_name* element cannot exceed *maximum_number* characters.

Explanation: The specified element contains too many characters.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI413S The XML structure of the *member_name* parameter metadata member is not valid. The *attribute_name* attribute in the *element_name* element is unknown.

Explanation: The specified attribute in the parameter metadata member is unknown.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI414S The content of the *member_name* parameter metadata member is not valid because the value of the *element_name* element is incorrect. The value is *value_name*.

Explanation: The specified value for an element in the parameter metadata member is not valid.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI415S The content of the *member_name* parameter metadata member is not valid because the value of the *attribute_name* attribute in the *element_name* element is incorrect. The value is *value_name*.

Explanation: The specified value for an attribute in the parameter metadata member is not valid.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI416S The content of the *member_name* parameter metadata member is not valid because the data type of the *element_name* element is incorrect. The value is *value_name*.

Explanation: The specified data type value for an element in the parameter metadata member is not valid.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI417S The content of the *member_name* parameter metadata member is not valid because the data type of the *attribute_name* attribute in the *element_name* element is incorrect. The value is *value_name*.

Explanation: The specified data type value for an attribute in the parameter metadata member is not valid.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI420S The XML structure of the *member_name* parameter metadata member is not valid. The *element_name* element is unknown for the overridden DB2 parameter.

Explanation:

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI421S The XML structure of the *member_name* parameter metadata member is not valid. The *element_name* element is unknown for the overridden LPAR parameter.

Explanation:

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI422S The XML structure of the *member_name* parameter metadata member is not valid. The *attribute_name* attribute in the *element_name* element is unknown for the overridden DB2 parameter.

Explanation:

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI423S The XML structure of the *member_name* parameter metadata member is not valid. The *attribute_name* attribute in the *element_name* element is unknown for the overridden LPAR parameter.

Explanation:

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI450S The *member_name* product parameter metadata member was not found in the *data_set_name* data set.

Explanation: Tools Customizer could not find the specified product parameter metadata member.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI510W The *data_set_name* data store data set does not exist.

Explanation: The specified data store data set does not exist.

System action: Processing continues.

User response: Ensure that the data store data set exists.

CCQI511S The *data_set_name* data store data set cannot be opened by using the *disposition_type* disposition.

Explanation: The specified data store data set could not be opened with the specified disposition.

System action: Processing continues.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI512S The *data_set_name* data store data set cannot be opened by using the *option-type* option.

Explanation: The specified data store data set was unable to be opened with the specified option.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI600W The XML structure of the *member_name* product customization parameter metadata member is not valid. The PL/I XML parser issued the following exception warning code: *code_number*.

Explanation: While determining if the product customization parameter metadata member is valid, the PL/I XML parser issued an exception warning code.

System action: Processing continues.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the warning.

CCQI601S The XML structure of the *member_name* product customization parameter metadata member is not valid. The PL/I XML parser issued the following exception error code: *code_number*.

Explanation: While determining if the product customization parameter metadata member is valid, the PL/I XML parser issued an exception error code.

System action: Processing continues.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the warning.

CCQI602S The XML structure of the *member_name* product customization parameter metadata member is not valid. The *element_name* element is unknown.

Explanation: The specified product customization parameter metadata member contains an unknown element.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI603S The XML structure of the *member_name* product customization parameter metadata member is not valid. Content is not allowed for the *element_name* element, but content was found.

Explanation: Content was found in an element that cannot contain content.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI604S The XML structure of the *member_name* product customization parameter metadata member is not valid. Content is required for the *element_name* element, but content was not found.

Explanation: The specified element does not contain required content.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI605S The XML structure of the *member_name* product customization parameter metadata member is not valid. The content length for the *element_name* element cannot exceed *maximum_number* characters.

Explanation: The specified element contains too many characters.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI606S The XML structure of the *member_name* product customization parameter metadata member is not valid. The *element_name* element cannot occur more than *maximum_number* times.

Explanation: The specified element occurs too many times in the product customization parameter metadata member.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI607S The XML structure of the *member_name* product customization parameter metadata member is not valid. The *element_name* element must occur at least *minimum_number* times.

Explanation: The specified element does not occur enough times in the product customization parameter metadata member.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI608S The XML structure of the *member_name* product customization parameter metadata member is not valid. The *attribute_name* attribute in the *element_name* element cannot occur more than *maximum_number* times.

Explanation: The specified attribute occurs too many times in the product customization parameter metadata member.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI609S The XML structure of the *member_name* product customization parameter metadata member is not valid. The *attribute_name* attribute in the *element_name* element must occur at least *minimum_number* times.

Explanation: The specified attribute does not occur enough times in the product customization parameter metadata member.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI610S The XML structure of the *member_name* product customization parameter metadata member is not valid. Content is not allowed for the *attribute_name* attribute in the *element_name* element, but content was found.

Explanation: Content was found in an element that cannot contain content.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI611S The XML structure of the *member_name* product customization parameter metadata member is not valid. Content is required for the *attribute_name* attribute in the *element_name* element, but content was not found.

Explanation: The specified attribute does not contain required content.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI612S The XML structure of the *member_name* product customization parameter metadata member is not valid. The content length for the *attribute_name* attribute in the *element_name* element cannot exceed *maximum_number* characters.

Explanation: The specified attribute contains too many characters.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI613S The XML structure of the *member_name* product customization parameter metadata member is not valid. The *attribute_name* attribute in the *element_name* element is unknown.

Explanation: The specified product customization parameter metadata member contains an unknown attribute.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI614S The XML structure of the *member_name* product customization parameter metadata member is not valid. The value of the *element_name* element is not valid. The value *value_name*.

Explanation: The specified value of the element is not a valid value.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI615S The XML structure of the *member_name* product customization parameter metadata member is not valid. The value of the *attribute_name* attribute for the *element_name* element is not valid. The value is *value_name*.

Explanation: The specified value of the attribute is not a valid value.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI616S The XML structure of the *member_name* product customization parameter metadata member is not valid. The data type of the *element_name* element is 'not valid. The value of the element is *value_name*.

Explanation: The specified data type is not a valid data type.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI617S The XML structure of the *member_name* product customization parameter metadata member is not valid. The data type of the *attribute_name* attribute for the *element_name* element is not valid. The value of the attribute is *value_name*.

Explanation: The specified data type is not a valid data type.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI650S The XML structure of the *member_name* product customization parameter metadata member is not valid. The following value of the *attribute_name* attribute in the *element_name* element already exists: *value_name*.

Explanation: The specified value for an attribute already exists.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI651S The XML structure of the *member_name* product customization parameter metadata member is not valid. The *parameter_name* parameter refers to the following section, which was not found in the *member_name* product customization parameter metadata member: *section-name*.

Explanation: The specified section is not in the product customization parameter metadata member.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI652S The *member_name* product customization metadata member not valid. The default length for the *element_name* parameter element exceeds the length of the parameter. The default length is *default_length*, and the specified length is *specified_length*. The default length will be truncated accordingly.

Explanation: The specified length cannot be shorter than the default length.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI653S The content of the *member_name* product customization parameter metadata member is not valid. The value of the *attribute_name* attribute in the *element_name* element is not valid. The value of the attribute is *value_name*.

Explanation: The specified value of the attribute is not a valid value.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI700W The XML structure of the *member_name* solution pack metadata member is not valid. The PL/I XML parser issued the following exception warning code: *code_number*.

Explanation: While determining if the specified solution pack metadata member is valid, the PL/I XML parser issued an exception warning code.

System action: Processing continues.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the warning.

CCQI701S The XML structure of the *member_name* solution pack metadata member is not valid. The PL/I XML parser issued the following exception error code: *code_number*.

Explanation: While determining if the specified solution pack metadata member is valid, the PL/I XML parser issued an exception error code.

System action: Processing stops.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the error.

CCQI702S The XML structure of the *member_name* solution pack metadata member is not valid. The *element_name* element is unknown.

Explanation: The specified solution pack metadata member contains an unknown element.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI703S The XML structure of the *member_name* solution pack metadata member is not valid. Content is not allowed for the *element_name* element, but content was found

Explanation: Content was found in an element that cannot contain content.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI704S The XML structure of the *member_name* solution pack metadata member is not valid. Content is required for the *element_name* element, but content was not found.

Explanation: The specified element does not contain required content.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI705S The XML structure of the *member_name* solution pack metadata member is not valid. The content length for the *element_name* element cannot exceed *maximum_number* characters.

Explanation: The specified element contains too many characters.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI706S The XML structure of the *member_name* solution pack metadata member is not valid. The *element_name* element cannot occur more than *maximum_number* times.

Explanation: The specified element occurs too many times.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI707S The XML structure of the *member_name* solution pack metadata member is not valid. The *element_name* element must occur at least *minimum_number* times.

Explanation: The specified element does not occur enough times.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI708S The XML structure of the *member_name* solution pack metadata member is not valid. The *attribute_name* attribute in the *element_name* element cannot occur more than *maximum_number* times.

Explanation: The specified attribute occurs too many times.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI709S The XML structure of the *member_name* solution pack metadata member is not valid. The *attribute_name* attribute in the *element_name* element must occur at least *minimum_number* times.

Explanation: The specified attribute does not occur enough times.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI710S The XML structure of the *member_name* solution pack metadata member is not valid. Content is not allowed for the *attribute_name* attribute in the *element_name* element, but content was found.

Explanation: The specified attribute cannot have content.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI711S The XML structure of the *member_name* solution pack metadata member is not valid. Content is required for the *attribute_name* attribute in the *element_name* element, but content was not found.

Explanation: The specified attribute is missing content.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI712S The XML structure of the *member_name* solution pack metadata member is not valid. The content length for the *attribute_name* attribute in the *element_name* element cannot exceed *maximum_number* characters.

Explanation: The specified attribute contains too many characters.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI713S The XML structure of the *member_name* solution pack metadata member is not valid. The *attribute_name* attribute in the *element_name* element is unknown.

Explanation: The specified attribute in the solution pack metadata member is unknown.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI714S The XML structure of the *member_name* solution pack metadata member is not valid because the value of the *element_name* element is incorrect. The value is *value_name*.

Explanation: The specified value of the element is not a valid value.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI715S The XML structure of the *member_name* solution pack metadata member is not valid because the value of the *attribute_name* attribute in the *element_name* element is incorrect. The value of the attribute is *value_name*.

Explanation: The specified value of the attribute is not a valid value.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI716S The XML structure of the *member_name* solution pack metadata member is not valid because the data type of the *element_name* element is incorrect. The value is *value_name*.

Explanation: The specified data type is not a valid data type.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI717S The XML structure of the *member_name* solution pack metadata member is not valid because the data type of the *attribute_name* attribute in the *element_name* element is incorrect. The value of the attribute is *value_name*.

Explanation: The specified data type is not a valid data type.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI720S The XML structure of the *member_name* solution pack metadata member is not valid. The msg element is required for the *component_name* component that is not customizable.

Explanation: The msg element is required for the specified component, which cannot be customized by using Tools Customizer.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI750S The solution pack metadata member was not found in the *library_name* metadata library.

Explanation: Tools Customizer could not find the solution pack metadata member in the specified library.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI751S The version in the *library_name* solution pack metadata library is different than the version in the *library_name* component metadata library. The name of the pack is *pack_name*, and the name of the component is *component_name*.

Explanation: The version in the solution pack metadata library does not match the version in the component metadata library.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI752S The release in the *library_name* solution pack metadata library is different than the release in the *library_name* component metadata library. The name of the pack is *pack_name*, and the name of the component is *component_name*.

Explanation: The release in the solution pack metadata library does not match the release in the component metadata library.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQI753S The modification level in the *library_name* solution pack metadata library is different than the modification level in the *library_name* component metadata library. The name of the pack is *pack_name*, and the name of the component is *component_name*.

Explanation: The modification level in the solution pack metadata library does not match the modification level in the component metadata library.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQM002E The *command_name* line command is not valid: .

Explanation: The specified line command is not valid.

System action: Processing continues.

User response: Specify a valid line command on the panel.

CCQO000W The XML structure of the *member_name* discover parameter metadata member is not valid. The PL/I XML parser issued the following exception warning code: *code_number*.

Explanation: While determining if the discover parameter metadata member is valid, the PL/I XML parser issued an exception warning code.

System action: Processing continues.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code.

CCQO001S The XML structure of the *member_name* discover parameter metadata member is not valid. The PL/I XML parser issued the following exception error code: *code_number*.

Explanation: While determining if the Discover metadata member is valid, the PL/I XML parser issued an exception error code.

System action: Processing stops.

User response: See the *Enterprise PL/I for z/OS Programming Guide* for more information about the exception warning code. Contact IBM Software Support.

CCQO002S The XML structure of the *member_name* discover parameter metadata member is not valid. The *element_name* element is unknown.

Explanation: The specified element in the discover parameter metadata member is unknown.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQO003S The XML structure of the *member_name* discover parameter metadata member is not valid. Content is not allowed for the *element_name* element, but content was found.

Explanation: The specified element cannot contain content.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQO004S The XML structure of the *member_name* discover parameter metadata member is not valid. Content is required for the *element_name* element, but content was not found.

Explanation: The specified element is missing required content.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQO005S The XML structure of the *member_name* discover parameter metadata member is not valid. The content length for the *element_name* element cannot exceed *maximum_number* characters.

Explanation: The specified element contains too many characters.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQO006S The XML structure of the *member_name* discover parameter metadata member is not valid. The *element_name* element cannot occur more than *maximum_number* times.

Explanation: The specified element occurs too many times.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQO007S The XML structure of the *member_name* discover parameter metadata member is not valid. The *element_name* element must occur at least *minimum_number* times.

Explanation: The specified element does not occur enough times.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQO008S The XML structure of the *member_name* discover parameter metadata member is not valid. The *attribute_name* attribute in the *element_name* element cannot occur more than *maximum_number* times.

Explanation: The specified attribute occurs too many times.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQO009S The XML structure of the *member_name* discover parameter metadata member is not valid. The *attribute_name* attribute in the *element_name* element must occur at least *minimum_number* times.

Explanation: The specified attribute does not occur enough times.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQO010S The XML structure of the *member_name* discover parameter metadata member is not valid. Content is not allowed for the *attribute_name* attribute in the *element_name* element, but content was found.

Explanation: The specified attribute cannot contain content.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQO011S The XML structure of the *member_name* discover parameter metadata member is not valid. Content is required for the *attribute_name* attribute in the *element_name* element, but content was not found.

Explanation: The specified attribute requires content.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQO012S The XML structure of the *member_name* discover parameter metadata member is not valid. The content length for the *attribute_name* attribute in the *element_name* element cannot exceed *maximum_number* characters.

Explanation: The specified attribute contains too many characters.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQO013S The XML structure of the *member_name* discover parameter metadata member is not valid. The *attribute_name* attribute in the *element_name* element is unknown.

Explanation: The specified attribute is unknown.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQO014S The content of the *member_name* discover parameter metadata member is not valid because the value of the *element_name* element is incorrect. The value is *value_name*.

Explanation: A The specified value for an element in the discover parameter metadata member is not valid.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQO015S The content of the *member_name* discover parameter metadata member is not valid because the value of the *attribute_name* attribute in the *element_name* element is incorrect. The value is *value_name*.

Explanation: The specified value for an attribute in the discover parameter metadata member is not valid.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQO016S The content of the *member_name* discover parameter metadata member is not valid because the data type of the *element_name* element is incorrect. The value is *value_name*.

Explanation: The specified data type value for an element in the discover parameter metadata member is not valid.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQO017S The content of the *member_name* product parameter metadata member is not valid because the data type of the *attribute_name* attribute in the *element_name* element is incorrect. The value is *value_name*.

Explanation: The specified data type value for an

attribute in the product parameter metadata member is not valid.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQO050S The *data_set_name* Discover REXX EXEC data set could not be initialized or was not found.

Explanation: Tools Customizer could not find or could not initialize the specified Discover REXX EXEC data set.

System action: Processing stops.

User response: Ensure that the Discover REXX EXEC is specified correctly.

CCQO051W The *data_sharing_group_ID* data sharing group ID cannot contain more than four characters.

Explanation: The specified data sharing group ID contains too many characters.

System action: Processing continues.

User response: Ensure that the specified data sharing group ID does not exceed four characters.

CCQO052S The *REXX_EXEC_name* Discover REXX EXEC was not found in the *data_set_name* Discover data set.

Explanation: Tools Customizer could not find the Discover REXX EXEC in the specified data set.

System action: Processing stops.

User response: Ensure that the Discover data set was specified correctly.

CCQO053W The *LPAR_name* LPAR name cannot contain more than eight characters.

Explanation: The specified LPAR name contains too many characters.

System action: Processing continues.

User response: Ensure that the specified LPAR name does not exceed eight characters.

CCQO054W The *subsystem_ID* DB2 SSID cannot contain more than four characters. The record was not processed.

Explanation: The specified DB2 SSID contains too many characters.

System action: Processing continues.

User response: Ensure that the specified DB2 SSID

does not exceed four characters.

CCQO055W The *parameter_name* DB2 group attach name parameter is in the *record_name* Discover record, but a DB2 group attach name was not specified. The record was not processed.

Explanation: The Discover record contains a data sharing group parameter, but a DB2 group attach name was not specified.

System action: Processing continues.

User response: Ensure that information is specified correctly on the Discover Customized Product Information panel.

CCQO056W The *parameter_name* DB2 parameter in the *record_name* Discover record did not have a DB2 group attach name or a DB2 SSID. The record was not processed.

Explanation: The Discover record did not have a DB2 group attach name or a DB2 subsystem ID in the DB2 parameter.

System action: Processing continues.

User response: Ensure that information is specified correctly on the Discover Customized Product Information panel.

CCQO057W The Discover EXEC could not find the *parameter_name* parameter in the metadata for the product to be customized. The record was not processed.

Explanation: The specified parameter could not be found in the metadata for the product to be customized.

System action: Processing continues.

User response: Ensure that information is specified correctly on the Discover Customized Product Information panel.

CCQO058W The *parameter_name* product parameter name in the *record_type* Discover record does not start with CCQ_LPR_, CCQ_DB2_, or CCQ_PRD_. The record was not processed.

Explanation: The parameter in the record does not start with CCQ_DB2_, CCQ_LPAR_, or CCQ_PRD_.

System action: Processing continues.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQO059W The *parameter_name* product parameter cannot contain more than 72 characters. The record was not processed.

Explanation: The specified product parameter contains too many characters.

System action: Processing continues.

User response: Ensure that the specified product parameter does not exceed 72 characters.

CCQO060W The *record_name* Discover record from the REXX EXEC output must start with the following record type: *record_type*. The record was not processed.

Explanation: A Discover record from the REXX EXEC output must start with the specified DB2 record type.

System action: Processing continues.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQO061I If you do not have a previously customized version of the product, do not run the Discover EXEC. Press END to go to the Customizer Workplace panel.

Explanation: This message is issued when you customize a product for the first time. It prompts you to use the Discover EXEC to discover data from a previous customization of the specified product.

System action: Processing continues.

User response:

Tip: Using the Discover EXEC saves time and reduces errors that can occur when parameters are specified manually. If you want to use the Discover EXEC, specify the required information on the Discover Customized Product Information panel. Otherwise, press End to continue without discovering data from a previous customization of the product.

CCQO062W The Discover EXEC could not find the following *parameter_name* parameter in the DB2 metadata. The record was not processed.

Explanation: The specified parameter is missing in the DB2 metadata.

System action: Processing continues.

User response: If this parameter is required, contact IBM Software Support.

CCQO064W The *Discover-record* Discover record did not have a parameter name. The record was not processed.

Explanation: A parameter name was missing in the Discover record.

System action: Processing continues.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQO065W The value for the *parameter_name* parameter is ignored because it has more than *maximum_number* characters, which is the maximum length that is defined in the metadata. The value is *parameter_value*.

Explanation: The specified value exceeded the maximum allowed length, which was defined in the metadata. Tools Customizer truncated the extra characters.

System action: Processing continues.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQO066W The *record_name* Discover record from the Discover REXX EXEC output does not have a parameter value. The record was not processed.

Explanation: The Discover record was missing a parameter value from the Discover EXEC output.

System action: Processing continues.

User response: Ensure that information was specified correctly on the Discover Customized Product Information panel.

CCQO067W The *parameter_name* parameter is defined in the metadata to support one value, but more than one value was found. The last value was used.

Explanation: The definition of the parameter in the metadata supports one value, but more than one value was specified. Only the last value was used.

System action: Processing continues.

User response: Ensure that information was specified correctly on the Discover Customized Product Information panel.

CCQO068W The value of the *parameter_name* parameter is ignored because the parameter is defined as *internal=true*. The value is *value_name*.

Explanation: The specified value of the parameter is

ignored because it is defined as *internal=true*.

System action: Processing continues.

User response: Ensure that information was specified correctly on the Discover Customized Product Information panel.

CCQO069W The Discover EXEC did not find the *parameter_name* parameter in the LPAR metadata. The record was not processed.

Explanation: The specified parameter is missing from the LPAR metadata.

System action: Processing continues.

User response: Ensure that information was specified correctly on the Discover Customized Product Information panel.

CCQO070W The *record_type* Discover record contains an incorrect delimiter between the Environment section and the Data section. The record was not processed.

Explanation: Tools Customizer found an incorrect delimiter between the Environment section and the Data section.

System action: None.

User response: No action is required.

CCQO071W The *member_name* member could not be found in the *data_set_name* Discover data set.

Explanation: Tools Customizer could not find the specified Discover data set.

System action: None.

User response: No action is required.

CCQO072S The *member_name* discover metadata member was not found in the *data_set_name* metadata data set.

Explanation: Tools Customizer could not find the specified metadata member in the data set.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQO073E The *member_name* discover metadata member is not valid because the default length for the *element_name* parameter element exceeds the length of the parameter. The default length is *default_length*, and the specified length is *specified_length*. The default length will be truncated accordingly.

Explanation: The default length for the specified parameter element is longer than the parameter.

System action: Processing continues.

User response: No action is required.

CCQO074S The content of the *member_name* discover metadata member is not valid. The value of the *attribute_name* attribute in the *element_name* element is not valid. The value of the attribute is *value_name*.

Explanation: The specified value is not valid.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQO075W The *configuration_ID* configuration ID in the *record_name* Discover record is incorrect. The record was not processed.

Explanation: The specified configuration ID is not correct.

System action: Processing continues.

User response: No action is required.

CCQO076W The *configuration_ID* configuration ID cannot contain more than *maximum_number* characters. The record was not processed.

Explanation: The specified configuration ID contains too many characters.

System action: Processing continues.

User response: No action is required.

CCQO077S The discover metadata member was not found in the *data_set_name* component data set that is part of the *data_set_name* pack.

Explanation: The discover metadata member was not found in the specified component data set.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQO080I *Product_name* does not support the Discover process.

Explanation: The specified product does not support the Discover process.

System action: None.

User response: No action is required.

CCQP000E The value of the *mode_name* DB2 mode is not valid for the *level_name* DB2 level.

Explanation: The specified DB2 mode is not valid for the DB2 level.

System action: Processing stops.

User response: Specify a valid DB2 mode for the DB2 level.

CCQP001E The value of the *mode_name* DB2 mode is missing.

Explanation: The specified DB2 mode is not defined.

System action: Processing stops.

User response: Specify a value for the DB2 mode.

CCQP002E The value of the *mode_name* DB2 level is missing.

Explanation: The specified DB2 level is not defined.

System action: Processing stops.

User response: Specify a value for the DB2 level.

CCQP003E The value of the *level_name* DB2 level is not valid.

Explanation: The specified DB2 level does not have a valid name.

System action: Processing stops.

User response: Specify a valid value for the DB2 level.

CCQP004S The *parameter_name* parameter does not exist in the CCQ\$\$DB2 DB2 parameter metadata member.

Explanation: The CCQ\$\$DB2 DB2 parameter metadata member does not contain the specified parameter.

System action: Processing stops.

User response: See Gathering diagnostic information. Contact IBM Software Support.

CCQP005E The value of the *subsystem_ID* DB2 SSID is missing.

Explanation: The specified DB2 SSID is not defined.

System action: Processing stops.

User response: Specify a valid value for the DB2 SSID.

CCQP006E The value of the *group_attach_name* DB2 group attach name is missing.

Explanation: The specified DB2 group attach name is not defined.

System action: Processing stops.

User response: Specify a valid DB2 group attach name.

CCQQ000E Specify a valid metadata library. Each qualifier of the library must start with an alphabetic character and must be 1-8 alphanumeric characters. The library name must be 1-44 characters.

Explanation: The metadata library was not specified in the correct format. The high-level qualifier must contain alphanumeric characters, and the first character cannot be numeric. The name cannot contain wildcard characters, such as asterisks (*) and percent signs (%).

System action: Tools Customizer prompts for the correct library name.

User response: Specify a library in the correct format. If the message was issued on the Specify the Metadata Library (CCQPHLQ) panel, specify the product metadata library. The name of this library is *hlq.SCKZDENU*.

Do not specify the Tools Customizer metadata library, which is *hlq.SCCQDENU*.

CCQQ001E The *data_set_name* data set name that was specified for the metadata library was not found.

Explanation: The data set does not exist, or the data set name was written in the incorrect format. The high-level qualifier must contain alphanumeric characters, and the first character cannot be numeric. The name cannot contain wildcard characters, such as asterisks (*) and percent signs (%).

System action: Tools Customizer prompts for the correct data set name.

User response: Specify a data set name in the correct format.

CCQQ002E The data set name that was specified for the *library_name* metadata library cannot be opened.

Explanation: Tools Customizer could not open the data set.

System action: Tools Customizer prompts for an available data set.

User response: Ensure that the specified data set is available for Tools Customizer to open it.

CCQQ003E The *data_set_name* data set name that was specified for the metadata sample library is not valid. The data set must be in the following format: *HLQ.SxxxSAMP*.

Explanation: The specified data set name was not specified in the correct format.

System action: None.

User response: Specify the data set name in the following format: *HLQ.SxxxSAMP*, where *xxx* is the three-character prefix for the product.

CCQQ004E The *data_set_name* data set is being used by another user. Try again when the data set is not being used.

Explanation: Another user is using the specified data set.

System action: None.

User response: Ensure that the specified data set is not being used.

CCQQ009E The *data_set_name* data set name that was specified for the metadata library is not valid because the data set is empty.

Explanation: The specified data set is empty.

System action: Tools Customizer prompts for an available data set.

User response: Ensure that the specified data set is available for Tools Customizer to open it.

CCQQ011E The *library_name* metadata library for the component that is part of the *library_name* pack was not found in the catalog. The name of the pack is *pack_name*, and the name of the component is *component_name*.

Explanation: The specified metadata library is not in the catalog.

System action: None.

User response: Specify another metadata library.

CCQQ012E The *library_name* metadata library for the component that is part of the *library_name* pack cannot be opened.

Explanation: The specified metadata library cannot be opened.

System action: None.

User response: Ensure that the name of the library is specified correctly.

CCQS000I Tools Customizer is being invoked for the first time or the previous ISPF session ended before Tools Customizer was exited. In both cases, the fields on this panel are populated with default values. Review these default values or specify new values to be used to customize products or packs.

Explanation: When you customize a stand-alone product or a solution pack for the first time, or when an ISPF session unexpectedly ends before the ISPF profile is saved, you must specify or review your Tools Customizer user settings.

System action: Processing stops.

User response: Review and accept the default settings, or specify new settings.

CCQS001E The following command is not valid:
command_name.

Explanation: The specified command is not a valid command on the panel.

System action: Processing stops.

User response: Specify a valid command.

CCQS002W The *data_set_name* Discover data set could not be found.

Explanation: Tools Customizer could not find the specified data set.

System action: The data set will be allocated, and processing continues.

User response: Ensure that the data set name is specified correctly because the data set will be allocated with this name after the values are saved.

CCQS003W The *data_set_name* Discover data set was not found so it was created.

Explanation: Tools Customizer could not find the specified data set.

System action: Processing continues.

User response: Ensure that the data set name is specified correctly.

CCQS004I The settings were saved.

Explanation: The settings that you changed were saved.

System action: Processing continues.

User response: No action is required.

CCQS006W The length of a qualifier for the *data_set_name* customization library data set exceeds 26 characters.

Explanation: The qualifier for the customization library data set is too long. The qualifier cannot exceed 26 characters.

System action: Processing continues.

User response: Specify a qualifier that is 26 characters or less.

CCQS007E The discover data set *data_set_name* could not be opened with the *option-type* option.

Explanation: The specified option could not open the Discover data set.

System action: None.

User response: Specify a data set to which you have WRITE access.

CCQS008E An error occurred while the *data_set_name* Discover data set was being created.

Explanation: While the specified data set was being created, an error occurred.

System action: Processing continues.

User response: Ensure that you have WRITE authority access to this data set.

CCQS010E The customization library qualifier is not valid.

Explanation: The customization library qualifier that was specified is not valid.

System action: None.

User response: Specify a valid qualifier for the customization library.

CCQS011E The group attach option is not valid.

Explanation: The group attach option that was specified is not valid.

System action: None.

User response: Specify a valid option for the group attach option.

CCQS012E The Tools Customizer metadata library is not valid.

Explanation: The metadata library that was specified is not a valid data set.

System action: None.

User response: Specify a valid data set for the metadata library.

CCQS013E The Discover data set is not valid.

Explanation: The Discover data set that was specified is not a valid data set.

System action: None.

User response: Specify a valid Discover data set.

CCQS014E The data store data set is not valid.

Explanation: The data set that was specified is not a valid data set.

System action: None.

User response: Specify a valid data store data set.

CCQS015E Tools Customizer is already running.

Explanation: A session of Tools Customizer is already running in your environment. Only one Tools Customizer session is allowed.

System action: None.

User response: The trace data set is being used. Free the trace data set, and start Tools Customizer again.

CCQS018E Information on the first line of the job card exceeds 57 characters.

Explanation: The first line of the job card can contain only 57 characters. This character limit includes a continuation character.

System action: Tools Customizer clears the first line of the job card.

User response: Specify information that does not exceed 57 characters on the first line of the job card.

CCQS019E The required trace data set, *data_set_name*, is currently not accessible.

Explanation: The trace data set must be accessible.

System action: Processing stops.

User response: Ensure that the trace data set is accessible.

CCQS020E An error occurred while the customization library data set was being created. ALTER authority on the high-level qualifier for the customization library data set is required.

Explanation: To create the customization library data set, ALTER authority on the specified high-level qualifier must be granted.

System action: None.

User response: Ensure that ALTER authority for the specified customization library data set is granted.

CCQS021E The value *value_name* in the field that contains the cursor position is not valid.

Explanation: The specified value is not valid.

System action: None.

User response: Specify a valid value.

CCQS022E An error occurred while the customization library data set was being opened. UPDATE authority on the high-level qualifier for the customization library data set is required.

Explanation: To open the customization library data set, UPDATE authority on the specified high-level qualifier must be granted.

System action: None.

User response: Ensure that UPDATE authority for the specified customization library data set is granted.

CCQS023E An error occurred while the customization library data set was being opened. UPDATE authority on the high-level qualifier for the customization library data set is required.

Explanation: To open the customization library data set, UPDATE authority on the specified high-level qualifier must be granted.

System action: None.

User response: Ensure that UPDATE authority for the specified customization library data set is granted, or specify a different high-level qualifier for the customization library data set on the Tools Customizer Settings panel.

CCQS024E An error occurred while the customization library data set was being created. ALTER authority on the high-level qualifier for the customization library data set is required.

Explanation: To create the customization library data set, ALTER authority on the specified high-level qualifier must be granted.

System action: None.

User response: Ensure that ALTER authority for the specified customization library data set is granted, or specify a different high-level qualifier for the

customization library data set on the Tools Customizer Settings panel.

CCQS030E The following command is not a valid CREATE statement: *command_statement*.

Explanation: The specified CREATE command statement is invalid because it contains blanks or alphabetic characters.

System action: Processing stops.

User response: Specify a valid CREATE command statement. The correct syntax is CREATE *nn*, where *nn* is 1 - 99.

CCQS031E The following command is not a valid CREATE statement: *command_statement*. The number that can be specified with the CREATE command is 1 - 99.

Explanation: The specified CREATE command statement is invalid because it contains either 0 or a number greater than 99.

System action: Processing stops.

User response: Specify a valid CREATE command statement. The correct syntax is CREATE *nn*, where *nn* is 1 - 99.

CCQT000I The product configuration ID *copied_configuration_ID* was successfully copied from *configuration_ID*.

Explanation: The specified configuration ID was copied.

System action: None.

User response: No action is required.

CCQT001E The *command_name* line command was specified more than once, which is not allowed.

Explanation: The specified line command cannot be specified more than one time.

System action: Processing stops.

User response: Specify the line command only once.

CCQT002E The *configuration_ID* configuration ID already exists. Specify a different configuration ID.

Explanation: The specified configuration ID exists.

System action: Processing stops.

User response: Ensure that the specified configuration ID is unique.

CCQT003I The product configuration ID *configuration_ID* was created.

Explanation: The specified configuration ID was created.

System action: None.

User response: No action is required.

CCQT004I The product configuration ID *configuration_ID* was removed.

Explanation: The specified configuration ID was removed.

System action: None.

User response: No action is required.

CCQT005E The product configuration ID *configuration_ID* is not valid. The product configuration ID cannot contain a colon (:).

Explanation: The specified configuration ID contains a colon (:), but a colon is not valid.

System action: Processing stops.

User response: Specify a configuration ID that does not contain a colon.

CCQT006E The *configuration_ID* configuration ID exists. Specify a different configuration ID.

Explanation: The specified configuration ID exists.

System action: Processing stops.

User response: Specify another configuration ID.

CCQT007E The *configuration_ID* configuration ID exists but was removed from the list of configurations. To use this configuration ID, you must restore it.

Explanation: The specified configuration ID exists but was removed from the list of available configuration.

System action: Processing stops.

User response: Specify another configuration ID. To restore the specified configuration ID, issue the CREATE command, and specify the same configuration ID again.

CCQT008E The *configuration_ID* configuration ID exceeds *maximum_number* characters.

Explanation: The specified configuration ID contains too many characters.

System action: Processing stops.

User response: Specify another configuration ID that does not exceed the maximum number of characters that was set by DB2 Cloning Tool.

CCQT010I Create request for *configuration_ID* configuration was cancelled by user.

Explanation: The request to create the specified configuration was canceled.

System action: Processing stops.

User response: No action is required.

CCQT011I The *configuration_ID* configuration was not copied.

Explanation: The specified configuration was not copied.

System action: Processing stops.

User response: No action is required.

CCQT012I The *configuration_ID* configuration was not removed.

Explanation: The specified configuration was not removed.

System action: Processing stops.

User response: No action is required.

CCQT013I None of the configurations were copied or removed. All of the previously selected configurations are deselected.

Explanation: The selected configurations were not copied or removed, and they are deselected.

System action: Processing stops.

User response: No action is required.

CCQT014E Specify Y or N and press Enter to continue, or press End to cancel.

Explanation: A function requires input.

System action: Processing stops.

User response: To continue, specify Y or N and press Enter. Otherwise, press End to cancel.

CCQT015E The *command_name* command is not allowed during the process of "Select" configuration line command.

Explanation: The specified command is not allowed while the line command for selecting configurations is processing.

System action: Processing stops.

User response: Remove the specified line command.

CCQT016I The *configuration_ID* configuration was not created

Explanation: The specified configuration was not created.

System action: Processing stops.

User response: No action is required.

CCQT017I The *configuration_ID* configuration was not copied.

Explanation: The specified configuration was not copied.

System action: Processing stops.

User response: No action is required.

CCQT018E Specify Y or N, and press Enter.

Explanation: A function requires input.

System action: Processing stops.

User response: To continue, specify Y or N, and press Enter.

CCQT019I The select *configuration_ID* configuration process ended.

Explanation: The select process for the specified configuration is finished.

System action: Processing stops.

User response: No action is required.

CCQT020E The *configuration_ID* configuration was not created because the data store was not accessible.

Explanation: The specified configuration was not created because the data store could not be accessed.

System action: Processing stops.

User response: Ensure that the data store is accessible and create the configuration again.

CCQT021E The *configuration_ID* configuration was not copied because the data store was not accessible.

Explanation: The specified configuration was not copied because the data store could not be accessed.

System action: Processing stops.

User response: Ensure that the data store is accessible and copy the configuration again.

CCQT025I The *configuration_ID* configuration was not updated.

Explanation: The specified configuration was not updated because the edit process was canceled.

System action: Processing stops.

User response: No action is required.

CCQT027I The product configuration was successfully updated.

Explanation: The configuration was updated.

System action: Processing continue.

User response: No action is required.

CCQX001S *Product_name* has already been customized by using values from *data_set_name* data store data set. Switch to the specified data store data set to continue customizing this product.

Explanation: The specified product was customized by using values from the specified data store data set.

DB2 Cloning Tool messages:

Use the information in these messages to help you diagnose and solve DB2 Cloning Tool problems.

CKZ002E Insufficient storage size. Region size of at least 30000 is required

Explanation: There is insufficient storage available.

User response: Contact your systems administrator to increase the region size to 30000.

CKZ003E Must be numeric

Explanation: An invalid value was entered.

User response: Enter a valid numeric value in the field.

CKZ004E Invalid Value - You must enter a "Y" or "N"

Explanation: An invalid value was entered.

User response: Enter a valid value as described in the message text.

CKZ005E Invalid Value - You must enter a "/", "Y", "N" or blank

Explanation: An invalid value was entered.

User response: Enter a valid value as described in the message text.

System action: Processing stops.

User response: Use the specified data store data set to continue customizing the product.

CCQX002S *component_name* has already been customized by using values from *data_set_name* data store data set. Switch to the specified data store data set to continue customizing this component.

Explanation: The specified component was customized by using values from the specified data store data set.

System action: Processing stops.

User response: Use the specified data store data set to continue customizing the component.

CCQX011I *Product_name* was not found.

Explanation: The specified product was not found.

System action: Processing stops.

User response: Specify another product.

CKZ006E Invalid Value - You must enter a "YES", "NO" or blank

Explanation: An invalid value was entered.

User response: Enter a valid value as described in the message text.

CKZ007E Multi selection is not available

Explanation: More than one element was selected from the list.

User response: Choose only one element from the list.

CKZ008E Invalid Value - You must enter a numeric value from *from value* to *to value*

Explanation: An invalid value was entered.

User response: Enter a value that is between the two values listed provided in the message text.

CKZ009E Invalid Value - You must enter a *valid value*

Explanation: An invalid value was entered.

User response: Enter a valid value as described in the message text.

CKZ010I No objects match the filter

Explanation: The list of objects is empty. There are no objects that meet the filter criteria.

User response: Change the filter values or add new objects to the list.

CKZ011I Operation completed successfully

Explanation: The operation successfully completed.

User response: No action is required.

CKZ012E The profile with given name already exists

Explanation: The name you provided for the profile already exists.

User response: Enter a different unique profile name.

CKZ013E The source member list does not match the target member list

Explanation: The number of selected sources does not equal the number of selected target subsystems.

User response: Provide a valid target member for each source member.

CKZ014E You do not have access rights to this profile

Explanation: An attempt was made to edit a profile that was created with no access rights by another user ID. Access to the profile is denied.

User response: Select a different profile to edit.

CKZ015E You only have rights to view this profile

Explanation: An attempt was made to edit a profile that was created with view only rights by another user ID. Editing the profile is not allowed.

User response: Use the V line command to view the profile.

CKZ016E Job Template Variable has invalid name

Explanation: An invalid Job Template Variable name was entered. The variable name entered duplicates a reserved variable name.

User response: Enter a unique Job Template Variable name.

CKZ017E You cannot specify the PARTLEVEL keyword with the RI keyword

Explanation: A value was entered in the Partlevel field when the RI field was set to YES. This combination is not allowed.

User response: Either change RI to NO or remove the value from the Partlevel field.

CKZ018E You must specify a valid DD for XML object definition on the DD specification panel

Explanation: An invalid or an empty value for the XML object definition DD name was entered.

User response: Enter a valid value for the XML object definition DD name on the DB2 tablespace clone DD Specification panel.

CKZ019E Error during data set creation for XML object definition

Explanation: An error occurred when creating the data set for XML object definition.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ020E The data set specified for XML object definition already exists, but is not a PDS

Explanation: The data set specified in the table space cloning DD definition panel already exists, but it is not a PDS.

User response: Return to the DB2 tablespace clone DD Specification and specify a PDS for the XML DD.

CKZ021E You should fill in a DD for all set DD Names

Explanation: Not all required DD names have been specified.

User response: Enter a DD for each required DD name.

CKZ022E Invalid Value - You must enter "SOURCE", "YES", or "NO"

Explanation: An invalid value was entered in the PRIMARY field.

User response: Enter SOURCE to specify that the primary BSDS will remain as the primary, as it is on the source subsystem. Enter YES to specify that this member will be the primary member. Enter NO to specify that this member will not be the primary member.

CKZ023E Invalid Value - You must enter "SOURCE" or "NO"

Explanation: An invalid value was entered in the PRIMARY field.

User response: Enter SOURCE to specify that the primary BSDS will remain as the primary, as it is on the source subsystem. Enter NO to specify that this member will not be the primary member.

CKZ024E No more than one entry can have PRIMARY set to "YES"

Explanation: YES was entered in the PRIMARY field for more than one subsystem. Only one subsystem can be specified as the primary member.

User response: Remove the duplicate value from one of the subsystems.

CKZ025E All entries cannot have PRIMARY set to "NO"

Explanation: One of the subsystems must be specified as the primary member.

User response: Enter YES or SOURCE next to one of the subsystems.

CKZ026E Please select surviving target member(s)

Explanation: A surviving target member must be selected.

User response: Select at least one surviving target member.

CKZ027E Invalid Value - Please enter a DB2 subsystem ID

Explanation: An invalid DB2 subsystem ID was entered.

User response: Specify a valid DB2 subsystem ID.

CKZ028E Invalid Value - Please enter a data set name

Explanation: An invalid data set name was entered.

User response: Enter a valid data set name in the field.

CKZ029E Invalid Value - Please enter a member name

Explanation: An invalid value was entered.

User response: Enter a valid member name in the field.

CKZ030E The DD Name *dd name* already exists

Explanation: The name you provided for the DD name already exists.

User response: Enter a different unique DD name.

CKZ031E Invalid Value - You must enter "IN" or "OUT"

Explanation: An invalid value was entered in the DIR field.

User response: Enter IN to specify that this DD is provided to ADRDSSU as input; enter OUT to specify that the DD is provided as output.

CKZ032E Invalid Value - You cannot use a reserved DD name

Explanation: A reserved DD name was entered.

User response: Enter a unique DD name that is not reserved.

CKZ033E Invalid Value - Please enter a valid LISTDEF name

Explanation: An invalid LISTDEF name was entered.

User response: Enter a valid LISTDEF name in the field.

CKZ034I Job generation was canceled

Explanation: The job generation was canceled by the user.

User response: No action is required.

CKZ035E Invalid Value - Please enter a valid control HLQ

Explanation: An invalid high level qualifier was entered.

User response: Enter a valid high level qualifier in the field.

CKZ036E Invalid Value - Please enter "CR", "DB", "TS", "IX", "IS", "TB", or "VC"

Explanation: An invalid OBJECT-TRANSLATE object type was entered

User response: Enter a valid OBJECT-TRANSLATE type in the field, as described in the message text.

CKZ037E Invalid Value - Please enter a Source Name

Explanation: A source name was not entered.

User response: Enter a source name.

CKZ038E **Invalid Value - Please enter a Target Name**

Explanation: A target name was not entered.

User response: Enter a target name.

CKZ039E **Invalid Value - Please enter a System VCAT**

Explanation: The system VCAT was not specified for subsystem, which is used as target only.

User response: Either enter a system VCAT for the subsystem, or change the subsystem so it is not a target only subsystem.

CKZ059E **Can't open *profile_version* version profile. The profile was saved with the latest version of the product. Please install all product updates or create a new profile.**

Explanation: You cannot open the profile saved with the latest version of the product.

User response: Please install all product updates or create a new profile.

CKZ101E **ISPF error : *error text***

Explanation: An ISPF error occurred and caused the displayed message to be generated.

User response: Correct the error and retry. If necessary, review the ISPF documentation to determine the cause of the error.

CKZ102E **Invalid Command - Please enter a valid command**

Explanation: An invalid command was entered in the Command or Option line. Valid commands are listed on the panel.

User response: Enter a valid command.

CKZ103E **Invalid Line Command - Please enter a valid line command**

Explanation: An invalid line command was entered. Valid line commands are listed at the top of the panel.

User response: Enter a valid line command.

CKZ104E **Invalid Selection - Please enter a valid option**

Explanation: An invalid option was entered.

User response: Enter a valid option.

CKZ104E **Invalid Selection - Please enter a valid option**

Explanation: An invalid value was entered

User response: Enter a valid value in the field.

CKZ106I **Move is pending. Enter A(After) or B(Before)**

Explanation: The M(Move) line command was entered but an A(After) or B(Before) command was not specified.

User response: Enter the A(After) or B(Before) line command, depending on the desired resulting order.

CKZ107E **Result not found**

Explanation: The desired element was not found.

User response: Verify that you correctly typed the desired element.

CKZ108I **No item selected**

Explanation: There was no element selected from the list.

User response: Select at least one item from the list.

CKZ200E *message text*

Explanation: An internal error of DB2 control file routine or VSAM data repository routine occurred.

User response: Consult the user guide for the explanation of error codes from those routines. If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support.

CKZ201E **The DB2 Cloning Tool repository does not exist**

Explanation: The base PDS is not a repository HLQ.

User response: Ensure the HLQ variable specified for the VSAM data repository in the CKZCLIST is correct. If unable to resolve the problem, contact IBM Software Support.

CKZ202E **The DB2 Cloning Tool DB2 control file does not exist**

Explanation: The base PDS is not a DB2 control file HLQ.

User response: Ensure the HLQ variable specified for the DB2 control file in the CKZCLIST is correct. If unable to resolve the problem, contact IBM Software Support.

CKZ301E Invalid data set name

Explanation: An invalid data set name was entered

User response: Enter a valid data set name in the field.

CKZ302E Invalid JCL library name

Explanation: An invalid JCL library name was entered.

User response: Ensure the JCL library name is correct. If unable to resolve the problem, contact IBM Software Support.

CKZ303E Cannot open file *filename*

Explanation: The file name listed in the message cannot be opened.

User response: If unable to determine the reason the file open failed, contact IBM Software Support. Have available the listing that contains this message.

CKZ304E Cannot get text from skeleton

Explanation: Text could not be read from a skeleton.

User response: If unable to determine the reason that text cannot be retrieved from skeleton, contact IBM Software Support. Have available the listing that contains this message.

CKZ305E Error during generation of COPY JCL

Explanation: An error occurred when generating the COPY JCL.

User response: If unable to determine the cause of the failure to generate COPY JCL, contact IBM Software Support. Have available the listing that contains this message.

CKZ306E Error during generation of RENAME JCL

Explanation: An error occurred when generating the RENAME JCL.

User response: If unable to determine the cause of the failure to generate RENAME JCL, contact IBM Software Support. Have available the listing that contains this message.

CKZ307E Error during generation of UPDATE JCL

Explanation: An error occurred when generating the UPDATE JCL.

User response: If unable to determine the cause of the failure to generate UPDATE JCL, contact IBM Software Support. Have available the listing that contains this message.

CKZ308E Error during generation of START JCL

Explanation: An error occurred when generating the START JCL.

User response: If unable to determine the cause of the failure to generate START JCL, contact IBM Software Support. Have available the listing that contains this message.

CKZ309E Error during generation of STOP JCL

Explanation: An error occurred when generating the STOP JCL.

User response: If unable to determine the cause of the failure to generate STOP JCL, contact IBM Software Support. Have available the listing that contains this message.

CKZ310E Error during generation of COPYCHECK JCL

Explanation: An error occurred when generating the COPYCHECK JCL.

User response: If unable to determine the cause of the failure to generate COPYCHECK JCL, contact IBM Software Support. Have available the listing that contains this message.

CKZ311E Error during generation of SQL JCL

Explanation: An error occurred when generating the SQL JCL.

User response: If unable to determine the cause of the failure to generate SQL JCL, contact IBM Software Support. Have available the listing that contains this message.

CKZ312E Error during generation of FIX JCL

Explanation: An error occurred when generating the FIX JCL.

User response: If unable to determine the cause of the failure to generate FIX JCL, contact IBM Software Support. Have available the listing that contains this message.

CKZ313E Error during generation of BCSCLEAN JCL

Explanation: An error occurred when generating the BCSCLEAN JCL.

User response: If unable to determine the cause of the failure to generate BCSCLEAN JCL, contact IBM Software Support. Have available the listing that contains this message.

CKZ314E Error during generation of SETLOG SUSPEND

Explanation: An error occurred when generating the SETLOG SUSPEND JCL.

User response: If unable to determine the cause of the failure to generate SETLOG SUSPEND JCL, contact IBM Software Support. Have available the listing that contains this message.

CKZ315E Error during generation of SETLOG RESUME JCL

Explanation: An error occurred when generating the SETLOG RESUME JCL.

User response: If unable to determine the cause of the failure to generate SETLOG RESUME JCL, contact IBM Software Support. Have available the listing that contains this message.

CKZ316E Error during generation of DB2RBLDBSDS JCL

Explanation: An error occurred when generating the DB2RBLDBSDS JCL.

User response: If unable to determine the cause of the failure to generate DB2RBLDBSDS JCL, contact IBM Software Support. Have available the listing that contains this message.

CKZ317E Error during generation of DB2LGRNXCLEAN JCL

Explanation: An error occurred when generating the DB2LGRNXCLEAN JCL.

User response: If unable to determine the cause of the failure to generate DB2LGRNXCLEAN JCL, contact IBM Software Support. Have available the listing that contains this message.

CKZ318E Error during generation of DB2XCFCLEAN JCL

Explanation: An error occurred when generating the DB2XCFCLEAN JCL.

User response: If unable to determine the cause of the failure to generate DB2XCFCLEAN JCL, contact IBM Software Support. Have available the listing that contains this message.

CKZ319I JCL was successfully generated

Explanation: The JCL for the profile was successfully generated.

User response: No action is required.

CKZ320E Error during data set creation for JCL

Explanation: An error occurred when creating the data set for the JCL.

User response: If unable to determine the reason the creating failed, contact IBM Software Support. Have available the listing that contains this message.

CKZ321E Error generating source JCL

Explanation: An error occurred when generating the source JCL.

User response: If unable to determine the cause of the failure to generate Source JCL, contact IBM Software Support. Have available the listing that contains this message.

CKZ322E Error generating target JCL

Explanation: An error occurred when generating the target JCL.

User response: If unable to determine the cause of the failure to generate target JCL, contact IBM Software Support. Have available the listing that contains this message.

CKZ323E Error generating TCPIP server JCL

Explanation: An error occurred when generating the TCP/IP server JCL.

User response: If unable to determine the cause of the failure to generate TCP/IP server JCL, contact IBM Software Support. Have available the listing that contains this message.

CKZ324E Error during generation of MASKDEF

Explanation: An error occurred when generating the MASKDEF JCL.

User response: If unable to determine the cause of the failure to generate MASKDEF JCL, contact IBM Software Support. Have available the listing that contains this message.

CKZ325E Error during generation of LISTDEF

Explanation: An error occurred when generating the LISTDEF JCL.

User response: If unable to determine the cause of the failure to generate LISTDEF JCL, contact IBM Software Support. Have available the listing that contains this message.

CKZ326E **Invalid Value - You must enter "YES" or "NO"**

Explanation: An invalid value was entered.

User response: Enter a valid value as described in the message text.

CKZ327E **You must specify special ZPARMS for DB2 subsystem *subsystem ID* before generating jobs**

Explanation: The ZPARMS field was not specified for this DB2 subsystem before job generation.

User response: Edit the specified DB2 subsystem under Administrator Functions on the main menu. Provide the ZPARMS member name for the subsystem.

CKZ328E **DB2 subsystem *subsystem ID* not found**

Explanation: The subsystem with the listed SSID is not found in the list of subsystems.

User response: Verify that you correctly entered the SSID, or select another DB2 subsystem.

CKZ329E **You must specify load library data sets for DB2 subsystem *subsystem ID* before generating jobs**

Explanation: The load library was not defined for the subsystem specified in the message.

User response: Edit the specified DB2 subsystem under Administrator Functions on the main menu. Provide at least one load library for the subsystem.

CKZ330E **File tailoring open returned a file tailoring already in progress condition**

Explanation: An attempt to perform file tailoring for utility customization failed. There was a file tailoring session already in progress. File tailoring sessions cannot be performed concurrently.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ331E **File tailoring open returned the output file already in use condition -- ENQ failed**

Explanation: An attempt to open the DB2 control data set failed with an ENQ error. The data set is already open for output.

User response: Verify that you are the only user attempting to access this file.

CKZ332E **File tailoring open returned the skeletal file or output file not allocated condition**

Explanation: An attempt to perform file tailoring failed because either the tailoring skeleton file or output file is not allocated.

User response: An attempt to perform file tailoring failed because either the tailoring skeleton file or output file is not allocated.

CKZ333E **File tailoring open returned a severe error condition**

Explanation: An attempt to perform file tailoring failed because a severe error condition was encountered on open.

User response: Verify that all required files are allocated and accessible prior to performing file tailoring.

CKZ334E **File tailoring open returned an unknown code -- severe error**

Explanation: An attempt to perform file tailoring failed because a severe error condition was encountered on open.

User response: Verify that all required files are allocated and accessible prior to performing file tailoring.

CKZ335E **File tailoring close returned a file not open condition -- severe error**

Explanation: An attempt to perform file tailoring failed because a File-Not-Open condition was encountered on close.

User response: Verify that all required files are allocated and accessible and that there are no other tailoring sessions running concurrently with your session.

CKZ336E **File tailoring close returned an output file in use condition**

Explanation: An attempt to perform file tailoring failed because an Output-File-In-Use condition was encountered on close.

User response: Verify that all required files are allocated and accessible and that there are no other tailoring sessions running concurrently with your session.

CKZ337E **File tailoring close returned a skeletal file or output file not allocated condition**

Explanation: An attempt to close file tailoring failed because either a tailoring skeleton file or output file was not allocated.

User response: Verify that all required files are allocated and accessible and that there are no other tailoring sessions running concurrently with your session.

CKZ338E **File tailoring close returned a severe error**

Explanation: An attempt to perform file tailoring failed because a severe error condition was encountered on close.

User response: Verify that all required files are allocated and accessible prior to performing file tailoring.

CKZ339E **File tailoring close returned an unknown code -- severe error**

Explanation: An attempt to perform file tailoring failed because a severe error condition was encountered on close.

User response: Verify that all required files are allocated and accessible prior to performing file tailoring.

CKZ340E **File tailoring close returned an output member exists in the output library and NOREPL was specified**

Explanation: An attempt to perform file tailoring failed because the close process could not replace the pre-existing tailored member in the output file.

User response: Change the output member name to a new name or ensure that the output library allows for member replacement.

CKZ341E **File tailoring include returned a skeleton does not exist condition**

Explanation: An attempt to perform file tailoring failed because the tailoring process could not locate a required tailoring skeleton.

User response: Ensure that all required files are allocated to perform file tailoring.

CKZ342E **File tailoring include returned a skeleton in use -- ENQ failed condition**

Explanation: An attempt to access a tailoring skeleton failed with an ENQ error (member in use).

User response: Verify that all required tailoring files are allocated and that there are no other tailoring sessions running concurrently.

CKZ343E **File tailoring include returned a data truncation or skeleton library or output file not allocated condition**

Explanation: An attempt to perform file tailoring failed because either the tailoring skeleton file or output file is not allocated.

User response: Verify that all required files are allocated prior to performing file tailoring.

CKZ344E **File tailoring include returned a severe error condition**

Explanation: An attempt to perform file tailoring failed because a severe error condition was encountered on an include operation.

User response: Verify that all required files are allocated and accessible prior to performing file tailoring.

CKZ345E **File tailoring include returned an unknown condition -- severe error**

Explanation: An attempt to perform file tailoring failed because a severe error condition was encountered on an include operation.

User response: Verify that all required files are allocated and accessible prior to performing file tailoring.

CKZ346E **Allocation Error - An error was encountered allocating the ISPF DD - Process not completed**

Explanation: An error was encountered allocating the ISPF DD.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ347E **Allocation Error - An error was encountered reading the ISPF DD - Process not completed**

Explanation: An error was encountered reading the ISPF DD.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ351W **Masking validation routine cannot be found. Check the user guide for correct syntax**

Explanation: The module CKZ00991 is not found.

User response: Check that the module CKZ00991 is available in ISPLLIB. Contact your system administrator to correct this error.

CKZ352E *message text*

Explanation: An internal error of masking routine occurred.

User response: Consult the user guide for the explanation of error codes from the masking routine. If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support.

CKZ353E **Unknown error of masking validation routine**

Explanation: An unknown error was encountered in the masking validation routine.

User response: Contact IBM Software Support.

CKZ354E **DDNAMEs *ddname1* and *ddname2* have the same data set name in Source DD Specification**

Explanation: The same data set name has been specified for different DDs.

User response: Change one of the data set names.

CKZ355E **DDNAMEs *ddname1* and *ddname2* have the same data set name in Target DD Specification**

Explanation: The same data set name has been specified for different DDs.

User response: Change one of the data set names.

CKZ356E **DDNAMEs *ddname1* and *ddname2* have the same data set name in TCPIP Server DD Specification**

Explanation: The same data set name has been specified for different DDs.

User response: Change one of the data set names.

CKZ360E **You must specify LOCAL-SSID for the SET command before generating jobs**

Explanation: A LOCAL-SSID was not specified for the SET command.

User response: Edit the profile and enter a

LOCAL-SSID in the field for the SET command before generating jobs.

CKZ361E **You must specify TARGET-DB2 SSID for the COPY command before generating jobs**

Explanation: A TARGET-DB2 SSID was not specified for the COPY command.

User response: Edit the profile and enter a TARGET-DB2 in the field for the COPY command before generating jobs.

CKZ362E **You must specify LOCAL-SSID for the TCPIP Server SET command before generating jobs**

Explanation: A LOCAL-SSID was not specified for the TCPIP Server SET command.

User response: Edit the profile and enter a LOCAL-SSID in the field for the TCPIP Server SET command before generating jobs.

CKZ363E **There is no data sharing group to satisfy the number of members for targets**

Explanation: There are no data sharing groups that can be used as targets, because none of the potential target data sharing groups have enough members.

User response: Add the required number of members into an existing target data sharing group or create the new target data sharing group with enough members to satisfy the selected source.

CKZ364E **There are not enough available members for targets in the target data sharing group**

Explanation: There are no data sharing groups that can be used as targets, because none of the potential target data sharing groups have enough members.

User response: Add more target member in the selected target data sharing group or select source members for cloning from another data sharing group

CKZ365E **You must specify System VCAT for DB2 subsystem *subsystem ID* before generating jobs**

Explanation: The system VCAT was not specified for the DB2 subsystem described in the message text.

User response: Enter a system VCAT for the DB2 subsystem described in the message text before generating jobs.

CKZ366E You must specify all table creators for the MASKDEF Command or the DEFAULT-SQLID for SET Command before generating jobs

Explanation: Either the MASKDEF command did not specify all table creators, or the DEFAULT-SQLID field for the SET command was not filled in.

User response: Either specify all table creators for MASKDEF Command, or specify DEFAULT-SQLID for the SET command.

CKZ367E Target DD *ddname* has different data set name from source. Please correct the target DD

Explanation: Different DDs were specified for same DD name on the DB2 tablespace clone DD Specification panel and the setup target DD cards panel.

User response: Change the target DD of the DD name shown in the message so that both DDs are the same in the DB2 tablespace clone DD Specification panel and in the Setup Target DD Cards panel.

CKZ368E You cannot use the same DDNAME *ddname1* in JOB-TEMPLATE and DD Specification

Explanation: The same DD names were defined in the JOB-TEMPLATE and the DD Specification.

User response: Change or remove one of the DD names.

CKZ369E You cannot use the same data set name in JOB-TEMPLATE and DDNAME *ddname1* of DD Specification

Explanation: The same data set names were defined in the JOB-TEMPLATE and one of the DD names in the DD Specification.

User response: Change one of data set names or remove a DD name from the DD Specification.

CKZ380E *symbol1* is not supported

Explanation: An invalid value was entered. This combination of field values is not allowed.

User response: Change the value of at least one field. If the object type is DATABASE or STOGROUP, changed the Object Specification Qualifier 1 field. Otherwise, change the value of either the Object Specification Qualifier 1 field or the Object Specification Qualifier 2 field.

CKZ384E Error on file *filename* I/O Handler: *error message*

Explanation: An error occurred that is documented in the message.

User response: Correct the error described in the message and retry.

CKZ410I DATA-MOVER PGM was changed to "SRCIMCPY". Therefore the following values of keywords were set: LA-ENABLE(Y), DATA-MASKING(N), SIM(N), SUBTASK-DATASET-EXTENSIONS(N), USE-RUNTIME REPOSITORY(N)

Explanation: DATA-MOVER PGM was set to SRCIMCPY. To clone from source image copies, log apply is required; therefore, LA-ENABLE for the LOG-APPLY command was set to YES. In addition, settings that are incompatible with cloning from source image copies were modified as follows:

- DATA-MASKING and SIM for the COPY command were set to NO.
- SUBTASK-DATASET-EXTENSIONS and USE-RUNTIME-REPOSITORY for the SET command were set to NO.

User response: No action is required.

CKZ411E Length of this string must be 12 or 20 hexadecimal characters

Explanation: The specified value is not valid.

User response: Enter a valid value.

CKZ412E You must specify *required_parameter* if END-POINT TYPE is set to *end_point_type*

Explanation: A required parameter was not specified. The required parameter is listed in the message.

User response: Enter the required parameter.

CKZ413E When DATA-MOVER PGM for the COPY Command is "SRCIMCPY", *parameter* cannot be enabled

Explanation: The parameter value that is listed in the message is not valid if DATA-MOVER PGM for the COPY command is set to SRCIMCPY.

User response: Change the value of either the parameter or the DATA-MOVER PGM field for the COPY command.

CKZ414E **The TARGET-JOB-INDEX-REBUILD-DDN value must be provided in the COPY command when REBUILD-INDEXES-EXECUTE is set to "YES" in the SET command**

Explanation: The TARGET-JOB-INDEX-REBUILD-DDN field for the COPY command is not set. This field is required if REBUILD-INDEXES-EXECUTE is set to YES in the SET command.

User response: Enter a value in the TARGET-JOB-INDEX-REBUILD-DDN field for the COPY command, or set the value of REBUILD-INDEXES-EXECUTE to NO for the Set command.

CKZ415E **Error during generation of RESTORE-FROM-DUMPTAPES JCL**

Explanation: An error occurred when generating RESTORE-FROM-DUMPTAPES JCL.

User response: If you are unable to determine the reason for the error from the associated z/OS messages, contact IBM Software Support. Have available the listing that contains these messages.

CKZ416E **Required parameter is not defined**

Explanation: A required parameter was not specified.

User response: Specify the required parameter and retry.

CKZ417E **Invalid Value - Enter either a numeric value from 1 to 16 or asterisk (*)**

Explanation: An invalid value was entered.

User response: Enter a valid value in the field.

CKZ418E **You must specify target volumes or target storage groups when using the RESTORE-FROM-DUMPTAPES command**

Explanation: Neither target volumes nor target storage groups were specified, but at least one of them is required.

User response: Specify either target volumes or target storage groups for the Source and Target Volume Pairing and retry.

CKZ419E **You must specify target storage groups if "Pair using Source Storage Group Names" field is equal to "YES"**

Explanation: Target storage groups were not specified, but they are required.

User response: Either specify target storage groups for

the Source and Target Volume Pairing or change the value of "Pair using Source Storage Group Names" field and specify target volumes.

CKZ430E **Invalid Value - Enter a valid *field_name* name**

Explanation: An invalid field was entered. The field in which the invalid name was entered is listed in the message text.

User response: Enter a valid value. The entered name must be alphanumeric or national characters; the first one can be alphabetic or national.

CKZ431E **Invalid Value - You must enter a "ALL", "LOB", "BASE", "XML" or blank**

Explanation: An invalid value was entered.

User response: Enter a valid value as described in the message text.

CKZ801E **Pgm: *program name* Stmt: *statement type* Type:**

Explanation: This message is used to convert SQL return code information into a text message. The data from the SQLCA is called using DSNTIAR and formatted into this message.

User response: Ensure that the qualifier value in the Run Table Space Cloning package binds job matches the creator name used in the Create Table Space Cloning log apply table job. Refer to the DB2 for z/OS: SQL Reference for your version of DB2 to resolve.

CKZ810E **Invalid CNUM parm. Valid parms are ON, OFF, or blank**

Explanation: CNUM was issued with an invalid parameter. Issuing CNUM with no parameter acts as an ON/OFF toggle. ON and OFF are the only parameters accepted. ON turns the CNUM display on. OFF turns the CNUM display off.

User response: Use a valid CNUM parameter (ON, OFF, or blank)

CKZ811E **Invalid COLS parm. Valid parms are ON, OFF, or blank**

Explanation: COLS was issued with an invalid parameter. Issuing COLS with no parameters acts as an ON/OFF toggle. ON and OFF are the only parameters accepted.

User response: Enter COLS ON or COLS OFF. COLS ON turns the COLS display on; COLS OFF turns the COLS display off.

CKZ812I **The FIND command requires specification of a target string**

Explanation: No parameters were specified with the FIND command. No match can be made unless you specify a string to find.

User response: Enter a FIND parameter.

CKZ813E **The RFIND key works only after a FIND character string is entered**

Explanation: A repeat FIND (RFIND) was issued before a FIND command was issued. You must issue FIND before RFIND will work.

User response: Issue FIND prior to attempting to issue RFIND.

CKZ814E **An unknown column *column name* was specified**

Explanation: The column you specified with the SORT command is not known.

User response: Verify that you correctly typed the name of the column or select another column.

CKZ815E **SORT is not supported for the specified column**

Explanation: The column you attempted to SORT is not supported as a column on which to sort.

User response: Refer to the sort columns listed on the Define Sort Columns panel for a list of valid columns on which the sort can be based and redefine the sort.

CKZ816E **Max Sort Columns exceeded. Sorting first 9 columns**

Explanation: More columns were selected for sorting than are supported. Nine columns can be selected. Under certain circumstances the limit is less than nine, due to internal constraints. For example, sorting a date field can be implemented by three sorts of partial column fields. In that case, the column would

User response: Specify the appropriate allowable maximum number of sort columns.

CKZ817E **Invalid column selection. Set cursor to valid column**

Explanation: An invalid column was selected.

User response: Set the cursor to a valid column.

CKZ818E **Invalid command parameters**

Explanation: Invalid command parameters were entered.

User response: Correct the command input and resubmit.

CKZ819E **Invalid place for moved column. Can't move source column to the new position**

Explanation: The source column cannot be moved to the new position.

User response: Correct the command input and resubmit.

CKZ820E **Not enough space for scrolling unfixed columns**

Explanation: Not enough space is available for scrolling unfixed columns

User response: Shrink the fixed area.

CKZ821E **Operation not valid for specified column**

Explanation: An invalid operation was entered.

User response: Enter a valid operation.

CKZ822E **Unable to hide fixed columns**

Explanation: Fixed columns cannot be hidden.

User response: Either make a selected column unfixed or select another column to hide.

CKZ823E **Invalid value entered for column size: non-numeric data**

Explanation: An invalid Cmd value was entered. This must be a number between the values in the MIN and MAX fields.

User response: Either remove the invalid number or enter a valid one.

CKZ824E **Invalid value entered for column size: out of range**

Explanation: An invalid Cmd value was entered. This must be a number between the values in the MIN and MAX fields. MIN is the smallest acceptable value. MAX is the largest acceptable value.

User response: Either remove the invalid number or enter a valid one.

CKZ825E SIZE is not supported for the specified column

Explanation: SIZE is not supported for the specified column. You cannot change the size of the column.

User response: You can change the size of another column, in which the minimum and maximum sizes are unequal.

CKZ00002E SUBTASK *subtask_number*, CLOSE ACB ERROR, R15=*return_code*, ERFLG=ACBERFLG

Explanation: This is an internal error.

User response: Call IBM Software Support.

CKZ00003I SUBTASK *subtask_number*, DB2 LDS CLOSE OK, DDN=*ddname*, DSN=*data_set_name*

Explanation: This message is informational.

User response: No action is required.

CKZ00004E COPY COMMAND PARSING ERROR, RC=*return_code*, RS=*reason_code*

Explanation: An error has been returned from the initialization parsing routine. This is a probable user error.

User response: Correct the input and resubmit the job.

CKZ00005I SUBTASK *subtask_number*, DB2 LDS ALLOCATION OK, *data_set_name*

Explanation: This message is informational. LDS is a VSAM linear data set.

User response: No action is required.

CKZ00006E SUBTASK *subtask_number*, DB2 COMMAND RESPONSE PARSING ERROR, RC=*return_code*, RS=*reason_code*

Explanation: A parsing error occurred while processing DB2command output. This is a probable internal error due to unexpected DB2 command output.

User response: Call IBM Software Support.

CKZ00007I SUBTASK *subtask_number*, DB2 LDS FREE OK, *data_set_name*

Explanation: This message is informational. LDS is a VSAM linear data set.

User response: No action is required.

CKZ00008I FOUND DATASET EXTENSION *data_set_name*

Explanation: This message is informational. A non-partitioned data set extension (other than A001) has been found in a z/OS catalog.

User response: No action is required.

CKZ00009E SUBTASK *subtask_number*, GENCB ACB ERROR, R15=*return_code*, R0=*reason_code*

Explanation: This is an internal error.

User response: Call IBM Software Support.

CKZ00010E SUBTASK *subtask_number*, GENCB EXL ERROR, R15=*return_code*, R0=*reason_code*

Explanation: This is an internal error. EXL is the VSAM exit list.

User response: Call IBM Software Support.

CKZ00011E SUBTASK *subtask_number*, GENCB RPL ERROR, R15=*return_code*, R0=*reason_code*

Explanation: This is an internal error.

User response: Call IBM Software Support.

CKZ00012E INTERNAL ERROR, RC=*return_code*, RS=*reason_code*, PLEASE CONTACT SUPPORT CENTER

Explanation: This is an internal error.

User response: Call IBM Software Support.

CKZ00013E REQUIRED KEYWORD, *command_keyword*, MISSING FROM COMMAND

Explanation: This is a user error.

User response: Supply the missing keyword.

CKZ00014W DB2_COMMAND_RESPONSE_WAIT PARM HAS AN INVALID VALUE, SET COMMAND IGNORED

Explanation: This is a user error.

User response: Specify a decimal number from 1 to 999999 as the number of seconds to wait for all DB2 spaces to stop and start. If the PARMLIB value for MAX_RC is set to 4, a default number of seconds will be used and the job will continue.

CKZ00015E TARGET-SSID *db2_subsystem* MUST BE FROM 1-4 CHARACTERS

Explanation: This is a user error.

User response: Specify a valid DB2 subsystem name.

CKZ00016E KEYWORD *command_keyword* HAS TOO MANY OPERANDS

Explanation: This is a user error.

User response: Specify the correct number for operands for the command.

CKZ00017E KEYWORD *keyword* HAS NO OPERANDS

Explanation: This is a user error.

User response: Specify the correct number for operands for the command.

CKZ00018I ?? SEGMENT HEADER BITS, FIRST BLOCK, ASSUMING LOB TS

Explanation: This is an informational message.

User response: No action is required.

CKZ00019I LOB TS SEGMENT HEADER BITS

Explanation: This is an informational message.

User response: No action is required.

CKZ00020E LOCK ALLOCATE ERROR, RS=*reason_code*

Explanation: This is an internal error.

User response: Call IBM Software Support.

CKZ00021E LOCK RELEASE ERROR, RS=*reason_code*

Explanation: This is an internal error.

User response: Call IBM Software Support.

CKZ00022E SUBTASK *subtask_number*, MODCB ERROR, R15=*return_code*, R0=*reason_code*

Explanation: This is an internal error.

User response: Call IBM Software Support.

CKZ00023W SUBTASK *subtask_number*, SSID *db2_subsystem*, CATALOG INCONSISTENCY, DATABASE *database_name* NOT FOUND

Explanation: This is a DB2 catalog error. The database that was previously found, could not be found.

User response: Call IBM Software Support if unable to resolve this catalog error.

CKZ00024E SUBTASK *subtask_number*, OPEN ACB ERROR, R15=*return_code*, ERFLG=ACBERFLG

Explanation: This may or may not be an internal error.

User response: Call IBM Software Support if unable to resolve this error.

CKZ00025I SUBTASK *subtask_number*, DB2 LDS OPEN OK, HI ALLOC RBA=*vsam_rba*, HI USED RBA=*vsam_rba* DDN=*ddname* DSN=*data_set_name*

Explanation: This message is informational.

User response: No action is required.

CKZ00026I OBJECT TRANSLATE CREATOR IN USE, SOURCE=*source_creator*, TARGET=*target_creator*

Explanation: This message is informational.

User response: No action is required.

CKZ00027I BEGIN PAGE PROC, OBJ: *object* SEQ#: SEQNR FLG:*flag_byte* DSN: *data_set_name*

Explanation: This message is informational.

User response: No action is required.

CKZ00028I DICTIONARY PAGE DB.TS REPLACEMENT, *target_database.target_table_space*, *source_database.source_table_space*

Explanation: This message is informational.

User response: No action is required.

CKZ00029E PARSE ERROR: FIRST TICK MUST BE AT THE START OF A TOKEN...

Explanation: This is a user error.

User response: Correct the input and resubmit the job.

CKZ00030E PARSE ERROR: TWO TICKS IN A ROW...

Explanation: This is a user error.

User response: Correct the input and resubmit the job.

CKZ00031E PARSE ERROR: EOF ENCOUNTERED WHILE IN A TICKED CLAUSE...

Explanation: This is a user error.

User response: Correct the input and resubmit the job.

CKZ00032E PARSE ERROR: ONE TICK BY ITSELF...

Explanation: This is a user error.

User response: Correct the input and resubmit the job.

CKZ00033W PAGE SET: READ
 ERRORS=*decimal_number_of_errors*,
 WRITE
 ERRORS=*decimal_number_of_errors*

Explanation: One or more errors were encountered while accessing the page set data sets in the target job.

User response: Treat these errors as if they are DB2 IO errors on the target spaces. Contact your systems programmer.

CKZ00034I Subtask *subtask_number*, Page Set:
 PGRK=*count_of_reads*,
 PGWK=*count_of_writes*,
 PTT=*count_of_points*,
 PGRW=*count_of_writes*,
 PTBK=*points_back_to_read_next_page* (V7)

Explanation: This message is informational.

User response: No action is required.

CKZ00035I HEADER PAGE CHANGES --- TBL
 OBID *old_table_object_id*
new_table_object_id, RECORD DESC
old_record_descriptor_object_id
new_record_descriptor_object_id, IX OBID
old_index_object_id new_index_object_id

Explanation: This message is informational.

User response: No action is required.

CKZ00036I RESET PGLOGRBA, WAS
old_table_LOGRBA

Explanation: This message is informational.

User response: No action is required.

CKZ00037I HEADER PAGE CHANGES ---
 DBID/PSID *old_dbid new_dbid, logrba*
 RESET, SSID *old_ssid new_ssid*, OBID
old_obid new_obid

Explanation: This message is informational.

User response: No action is required.

CKZ00038I SG/VCAT CHANGES --- SG: *sss ttt*,
 VCAT: *vvv www*

Explanation: This message is informational. *sss* = old SG *ttt* = new SG *vvv* = old VCAT *www* = new VCAT

User response: No action is required.

CKZ00039I INDEX DATA PAGE, OBID CHANGE
old_obid new_obid, PAGE OFFSET
offset_value

Explanation: This message is informational.

User response: No action is required.

CKZ00040I BEGIN PROCESSING INDEX
 DIRECTORY PAGE

Explanation: This message is informational.

User response: No action is required.

CKZ00041I UNFORMATTED INDEX PAGE,
 DISCARDED

Explanation: This message is informational.

User response: No action is required.

CKZ00042I BEGIN PROCESSING INDEX SPACE
 MAP

Explanation: This message is informational.

User response: No action is required.

CKZ00043I BEGIN PROCESSING INDEX DATA
 PAGE, IPFLAGS=*flags_in_the_page_header*

Explanation: This message is informational.

User response: No action is required.

CKZ00044I LOB DATA PAGE

Explanation: This message is informational.

User response: No action is required.

CKZ00045I LOB HIGH LEVEL SPACE MAP PAGE

Explanation: This message is informational.

User response: No action is required.

CKZ00046I LOB LOW LEVEL SPACE MAP PAGE

Explanation: This message is informational.

User response: No action is required.

CKZ00047I LOB MAP PAGE

Explanation: This message is informational.

User response: No action is required.

CKZ00048I UNIDENTIFIED LOB PAGE

Explanation: This message is informational.

User response: No action is required.

**CKZ00049I REPLACE PAGE SET DATA PAGE MAP
ID, *old_map_id new_map_id*, PAGE
OFFSET *page_offset***

Explanation: This message is informational.

User response: No action is required.

**CKZ00050I HEADER PAGE,
PGCOMB=*flags_in_pageset_header*,
FLAGS=*flags_in_pageset_header*,
HPGFLAGS=*flags_in_pageset_header***

Explanation: This message is informational.

User response: No action is required.

**CKZ00051I HEADER PAGE, LARGE,
PGCOMB=*flags_in_pageset_header*,
FLAGS=*flags_in_pageset_header*,
HPGFLAGS=*flags_in_pageset_header***

Explanation: This message is informational.

User response: No action is required.

**CKZ00052I BEGIN PROCESSING DATA PAGE,
#MAPIDS: *number_of_map_ids***

Explanation: This message is informational.

User response: No action is required.

**CKZ00053I BEGIN PROCESSING SEGMENTED
SPACE MAP WITH SEGSIZE
*segment_size***

Explanation: This message is informational.

User response: No action is required.

**CKZ00054I SPACE MAP CHANGES --- OBID
old_map_id new_map_id, PAGE OFFSET
*page_offset***

Explanation: This message is informational.

User response: No action is required.

**CKZ00055I SINGLE TABLE OBID CHANGE,
*old_obid new_obid***

Explanation: This message is informational.

User response: No action is required.

**CKZ00056W UNEXPECTED FLAGS=*flags*, PAGE
BYPASSED**

Explanation: This message is informational.

User response: If the target job runs with no errors and the target objects are accessible, ignore this error, otherwise call IBM Software Support.

**CKZ00057I NON HEADER PAGE, RESET
PGLOGRBA, WAS *old_rba***

Explanation: This message is informational.

User response: No action is required.

**CKZ00058I NO MAP TRANSLATE REQUIRED
FOR OBID *current_obid*, ROW OFFSET
*offset_to_start_of_row***

Explanation: This message is informational.

User response: No action is required.

**CKZ00059I ZERO PAGE NUMBER, NO
PROCESSING**

Explanation: This message is informational.

User response: No action is required.

CKZ00060E QUEUE READ ERROR, RS=*reason_code*

Explanation: This is an internal error.

User response: Call IBM Software Support.

CKZ00061E QUEUE WRITE ERROR, RS=*reason_code*

Explanation: This is an internal error.

User response: Call IBM Software Support.

**CKZ00062E INDEX *index_creator.index_name* IN LIST
listdef_list_name SPECIFIED RI, BUT NO
TABLE COULD BE FOUND**

Explanation: This is a probable DB2 catalog inconsistency.

User response: Resolve the inconsistency and resubmit the job.

CKZ00063E SUBTASK *subtask_number*, SHOWCB
ERROR, R15=*return_code*, R0=*reason_code*

Explanation: This is an internal error.

User response: Call IBM Software Support.

CKZ00064E SUBTASK *subtask_number*, *access_type*,
operation, RC=*return_code*, RS=*reason_code*,
ALLOCATION FAILED

Explanation: This is a probable system error. Unable to dynamically allocate a file using SVC99.

User response: Ensure that the file exists and is not allocated to DB2 or another job. Call IBM Software Support if unable to resolve this error.

CKZ00065E SUBTASK *subtask_number*, *access_type*,
operation, RC=*return_code*, RS=*reason_code*,
FREE FAILED

Explanation: This is a probable internal error. Unable to dynamically free a file using SVC99.

User response: Call IBM Software Support.

CKZ00066E SUBTASK *subtask_number*, *access_type*,
operation, RC=*return_code*, RS=*reason_code*,
ALLOCATION FAILED, ERROR
STRING = *error_string*

Explanation: This is a probable system error. Unable to dynamically allocate a file using SVC99.

User response: Ensure the file exists and is not allocated to DB2 or another job. Call IBM Software Support if unable to resolve this error.

CKZ00067I TCPIP SOCKET CLOSED, IP *ipaddress*

Explanation: This message is informational.

User response: No action is required.

CKZ00068W SUBTASK *subtask_number*, NO TABLES
FOUND IN TARGET TABLESPACE
database_name.table_space_name

Explanation: This is a probable user error.

User response: Add the table objects and resubmit the job.

CKZ00069I SOCKET CALL SUCCESSFUL,
SOCKET *socket_number*

Explanation: This message is informational.

User response: No action is required.

CKZ00070I TCPIP SOCKET CLOSED, IP *ipaddress*, -
SOCKET *socket_number*, PACKETS
RECEIVED *decimal_number*, PACKETS
SENT *decimal_number*

Explanation: This message is informational.

User response: No action is required.

CKZ00071I INITAPI CALL SUCCESSFUL, TCP
started_task_of_tcpip

Explanation: This message is informational.

User response: No action is required.

CKZ00072I RECEIVE *decimal_number_of_bytes_*
received_without_error BYTES, OK

Explanation: This message is informational.

User response: No action is required.

CKZ00073I SEND
decimal_number_of_bytes_sent_without_error
BYTES, OK

Explanation: This message is informational.

User response: No action is required.

CKZ00074E UNABLE TO LOAD TCP Module
FROM EZA LIBS TO EXECUTE
EZASMI_macro_that_failed EZASMI
MACRO

Explanation: This is a TCPIP environment error. Cannot load TCPIP module EZASOH03.

User response: Contact your system programmer to give DB2 Cloning Tool Table Space Cloning access to the TCPIP load library SEZALOAD.

CKZ00075E UNABLE TO ALLOCATE SOCKET,
RETCODE=*return_code*,
ERRNO=*tcpip_ERRNO_return_value*

Explanation: This is a TCPIP environment error.

User response: Contact your system programmer.

CKZ00076E UNABLE TO BIND TO SOCKET WITH
PORT *tcpip_port_number*,
RETCODE=*return_code*,
ERRNO=*tcpip_ERRNO_return_value*

Explanation: This is a TCPIP environment error. The port specified may be in use by another application.

User response: Specify another port number. Ensure the port number of the TCPIP client source job matches the port number of the TCPIP server job.

CKZ00077E CLOSE ERROR, RETCODE=*return_code*,
ERRNO=*tcPIP_ERRNO_return_value*

Explanation: This is a probable TCPIP error.

User response: Contact your system programmer.

CKZ00078E UNABLE TO CONNECT TO TCP
name_of_tcpip_address_space,
RETCODE=*return_code*,
ERRNO=*tcPIP_ERRNO_return_value*

Explanation: This is a user error or TCPIP error.

User response: Specify the correct TCPIP started task name and restart the job. If not resolved, contact your system programmer.

CKZ00079E RECEIVE DATA LENGTH ERROR,
EXPECTING *number_of_bytes_expected*,
RECEIVED *number_of_bytes_received*

Explanation: This is a probable TCPIP error.

User response: Contact your system programmer.

CKZ00080E RECEIVE DATA ERROR,
RETCODE=*return_code*,
ERRNO=*tcPIP_ERRNO_return_value*

Explanation: This is a probable TCPIP error.

User response: Contact your system programmer.

CKZ00081E SEND DATA ERROR,
RETCODE=*return_code*,
ERRNO=*tcPIP_ERRNO_return_value*

Explanation: This is a probable TCPIP error.

User response: Contact your system programmer.

CKZ00082E BUFFER ALLOCATION ERROR,
RC=*return_code_from_getmain*

Explanation: Unable to allocate virtual storage.

User response: Call IBM Software Support if unable to resolve this error by increasing the REGION size.

CKZ00083E SUBTASK *subtask_number*, BUFFER
ALLOCATION ERROR,
RC=*return_code_from_getmain*

Explanation: Unable to allocate virtual storage.

User response: Call IBM Software Support if unable to resolve this error by increasing the REGION size.

CKZ00084E SUBTASK *subtask_number*, BUFFER
RELEASE ERROR,
RC=*return_code_from_getmain*

Explanation: This is an internal error.

User response: Call IBM Software Support.

CKZ00085E SUBTASK *subtask_number*, VSAM GET
ERROR, R15=*return_code*,
RPLFDBK=*field_from_vsam_RPL*,
DSN=*data_set_name*

Explanation: This may be an IO error.

User response: Call IBM Software Support if unable to resolve this error.

CKZ00086W SUBTASK *subtask_number*, VSAM GET
Error, R15=*return_code*,
RPLFDBK=*field_from_vsam_RPL*,
DSN=*data_set_name*, DATASET
EXTENSION IS EMPTY

Explanation: This may be an IO error.

User response: Call IBM Software Support if unable to resolve this error.

CKZ00087E Subtask *subtask_number*, point error,
R15=*vsam_return_code*,
RPLFDBK=*vsam_feedback_code*,
DD=*ddname*

Explanation: An error occurred during VSAM processing.

User response: Call IBM Software Support if unable to resolve this error.

CKZ00088E SUBTASK *subtask_number*, VSAM PUT
ERROR, R15=*return_code*,
RPLFD=*field_from_vsam_RPL*

Explanation: This may be an IO error.

User response: Call IBM Software Support if unable to resolve this error.

CKZ00089E SUBTASK *subtask_number*, NO ENTRY
IN SYSIBM.SYSTABLES FOR BASE
TABLE
db2_subsystem.table_creator.table_name AS
POINTED TO BY
SYSIBM.SYSAUXRELS

Explanation: This is a DB2 catalog inconsistency.

User response: Resolve the inconsistency and resubmit the job.

CKZ00090E CKZIN OPERAND *operand_in_CKZIN*,
BAD LENGTH

Explanation: This is a user error.

User response: Correct the error and resubmit the job.

CKZ00091E UNRECOGNIZED CKZIN COMMAND,
command_name

Explanation: This is a user error.

User response: Correct the error and resubmit the job.

CKZ00092I DICTIONARY PAGE, NOT HEADER,
IGNORED

Explanation: This message is informational.

User response: No action is required.

CKZ00093I SUBTASK *subtask_number*, TARGET SS
db2_subsystem, OBJECT NOT IN DB2
CATALOG *object_type*
object_creator.object_name

Explanation: This message is informational.

User response: No action is required.

CKZ00094W TABLE *table_creator.table_name* AND
COLUMN *column_name*, DEFAULT
COLUMN=
col_default_from_SYSCOLUMNS, BUT NO
ENTRY FOUND IN
SYSIBM.SYSSEQUENCESDEP

Explanation: This is a DB2 catalog inconsistency. The column indicated it was an identity column, but it was not found in the identity column catalog table.

User response: Resolve the inconsistency and resubmit the job.

CKZ00095W TABLE *table_creator.table_name* AND
COLUMN *column_name*, DEFAULT
COLUMN =
col_default_from_SYSCOLUMNS, BUT NO
ENTRY FOUND IN
SYSIBM.SYSSEQUENCES FOR ID =
BSEQUENCEID_from_
SYSSEQUENCESDEP

Explanation: This is a DB2 catalog inconsistency. The column indicated it was an identity column, but it was not found in the identity column catalog table.

User response: Resolve the inconsistency and resubmit the job.

CKZ00096E *parameter_name* MUST BE SPECIFIED,
THERE IS NO DEFAULT

Explanation: This is a user error.

User response: Correct the error and resubmit the job.

CKZ00097E DATASET *data_set_name* HAS AN
INVALID PARTITION NUMBER

Explanation: This is a user error. The partition number specified is not within the range supported by DB2.

User response: Correct the error and resubmit the job.

CKZ00098I NON SEGMENTED SPACE MAP
PAGE, NO OBIDS TO TRANSLATE

Explanation: This message is informational.

User response: No action is required.

CKZ00099E SUBTASK *subtask_number*, NO ENTRY
IN SYSIBM.SYSTABLES FOR INDEX
db2_subsystem.index_creator.index_name

Explanation: This is a DB2 catalog inconsistency.

User response: Resolve the inconsistency and resubmit the job.

CKZ00101E SUBTASK *nn*, UNABLE TO CONNECT
TO TARGET SUBSYSTEM

Explanation: This is a DB2 or user error. *nn* = subtask number

User response: If the subsystem name is correct, determine why the connection fails. If the subsystem name is incorrect, correct the parameter and resubmit the job.

CKZ00102W SUBTASK *nn*, UNABLE TO READ
TARGET CATALOG, RC=*rrr*, RS=*sss*
OBJECT=*tt ccc.nnn*

Explanation: This may be a DB2 SQL error. *nn* = subtask number *rrr* = return code *sss* = reason code *tt* = object type *ccc* = object creator *nnn* = object name

User response: Correct the DB2 problem and resubmit the job.

CKZ00103W TABLE *ccc.nnn* REFERENCED
TABLESPACE *ddd.ooo*, BUT THE
TABLESPACE NOT FOUND IN
CATALOG.

Explanation: This is a DB2 catalog inconsistency. *ccc* = table creator *nnn* = table name *ddd* = database name *ooo* = tablespace name

User response: Resolve the inconsistency and resubmit the job.

CKZ00104E INVALID FORMAT, ssss

Explanation: This is a user error. ssss = string with the parameter error

User response: Correct the error and resubmit the job.

CKZ00105E SYNCDB2 INVALID XLATE
COMMAND, LENGTH=0

Explanation: This is a user error.

User response: Correct the error and resubmit the job.

CKZ00106E SUBTASK nn, NO ENTRY IN
SYSIBM.SYSAUXRELS FOR AUX TBL
ssss ccc.nnn

Explanation: This is a DB2 catalog inconsistency. nn= subtask number ssss = DB2 subsystem name ccc = aux table creator nnn = aux table name

User response: Resolve the inconsistency and resubmit the job.

CKZ00107I COPY ccc is = vvv

Explanation: This indicates the value of the a COPY Y/N command. ccc = copy command vvv = value

User response: No action is required.

CKZ00108E FAILED PRODUCT AUTHORIZATION

Explanation: The license in Parmlib is not valid.

User response: Call IBM Software Support if the license is correct in Parmlib.

CKZ00109E SUBTASK nn, SSID ssss, CATALOG
INCONSISTENCY, TABLE ccc.nnn NOT
FOUND

Explanation: This is a probable DB2 catalog error. nn= subtask number ssss = DB2 subsystem name ccc = table creator nnn = table name

User response: Call IBM Software Support if unable to resolve this catalog error.

CKZ00110E SUBTASK nn, SSID ssss, CATALOG
INCONSISTENCY, TABLESPACE
ddd.ttt NOT FOUND

Explanation: This is a probable DB2 catalog error. nn= subtask number ssss = DB2 subsystem name ddd = database name ttt = tablespace name

User response: Call IBM Software Support if unable to resolve this catalog error.

CKZ00111E SUBTASK nn, SSID ssss, CATALOG
INCONSISTENCY, nnn TABLES NOT
FOUND IN TABLESPACE ddd.ttt

Explanation: This is a probable DB2 catalog error. nn= subtask number ssss = DB2 subsystem name nnn = correct number of tables ddd = database name ttt = tablespace name

User response: Call IBM Software Support if unable to resolve this catalog error.

CKZ00112I TCPIP SOCKET CLOSED, IP
V6IPADDR

Explanation: This message is informational.

User response: No action is required.

CKZ00113I TCPIP SOCKET CLOSED, IP
V6IPADDR, - SOCKET ddd, PACKETS
RECEIVED eee, PACKETS SENT fff

Explanation: This message is informational. ddd = socket number eee = decimal number fff = decimal number

User response: No action is required.

CKZ00114W Subtask nn, SSID ssss, Catalog
Inconsistency, Base/Clone Pair not
Found for ccc.ttt

Explanation: This message is a warning. nn = subtask number ssss = DB2 subsystem name ccc = missing table creator ttt = missing table name

User response: Call IBM Software Support if unable to resolve this catalog error.

CKZ00116E Subtask nn, Unable to Load Module:
mmm

Explanation: This message is an error. nn = subtask number mmm = the module name that could not be loaded

User response: Ensure the module is available to DB2 Cloning Tool using STEPLIB, LPA, link library, etc.

CKZ00117E PARSE ERROR: UNBALANCED
COMMENTS, SOME DATA MAY BE
IGNORED

Explanation: This is a user error.

User response: Correct the input and resubmit the job.

CKZ00118I SUBTASK nn, OFFSET = ooo, XML TYPE HEADER =hhh

Explanation: This message is informational. nn = subtask number ooo = offset in the page being processed hhh = header contents in hex

User response: No action is required.

CKZ00119I SUBTASK nn, XML TYPE BLOCK hhh

Explanation: This message is informational. nn = subtask number hhh = contents in hex

User response: No action is required.

CKZ00120I SUBTASK nn, REPLACED SOURCE STRING ID hhh WITH TARGET STRING ID iii

Explanation: This message is informational. The target page has been updated with the new target string ID. nn = subtask number hhh = source string id in hex iii = target string id in hex

User response: No action is required.

CKZ00121E Subtask nn, Can?t Find Source String ID hhh, DSN(dsname), SEQ# seqnr, RC=rrr, RS=sss

Explanation: This message is an error. It might be DB2 or a program logic error, or could be the result of using an incorrect XMLSTRINGS DD. nn = subtask number hhh = source string id in hex rrr = return code sss = reason code

User response: Call IBM Software Support if unable to resolve this error.

CKZ00122E SUBTASK nn, CAN'T FIND XML DOC TYPE, hhh, DSN(dsname), SEQ# seqnr, RC=rrr, RS=sss

Explanation: This message is an error. It might be DB2 or a program logic error. Each XML table space should start with a DOCUMENT type ID. nn = subtask number hhh = source string ID in hex dsname = data set name seqnr = sequence number rrr = return code sss = reason code

User response: To determine if this XML space is bad, do a SELECT on the XML column on the source subsystem. If the SELECT runs with no errors, this is probably a DB2 Cloning Tool Table Space Cloning problem. Call IBM Software Support if unable to resolve this error.

CKZ00123E SUBTASK nn, UNABLE TO PROCESS XML FORMAT, DSN(dsname), SEQ# seqnr, RC=rrr, RS=sss

Explanation: This message is an error. It might be DB2 or a program logic error. XML node type is not recognized. nn = subtask number dsname = data set name seqnr = sequence number rrr = return code sss = reason code

User response: To determine if this XML space is bad, do a SELECT on the XML column on the source subsystem. If the SELECT runs with no errors, this is probably a DB2 Cloning Tool Table Space Cloning problem. Call IBM Software Support if unable to resolve this error.

CKZ00124I SUBTASK nn, SKIPPING SPACE MAP ENTRY, DSN(dsname), SEQ# seqnr, SMAP mmm

Explanation: This message is informational. nn = subtask number dsname = data set name seqnr = sequence number mmm = 2 byte space map entry

User response: No action is required.

CKZ00132I SUBTASK nn, XML CREATE STRING DATABASE ALREADY EXISTS,PROCESSING CONTINUES

Explanation: This message is informational. nn = subtask number

User response: No action is required.

CKZ00134E Subtask *subtask_number*, SSID *subsystem_ID*, catalog inconsistency, *object_type*, *object_qualifier_1.object_qualifier_2* not found

Explanation: This error occurs when rereading the DB2 catalog get columns required for DDL generation.

User response: Call IBM Software Support if unable to resolve this error.

CKZ00135E Subtask *subtask_number*, SSID *subsystem_ID*, Catalog Inconsistency, *object_type*, *object_qualifier_1.object_qualifier_2* *secondary_object_type* *secondary_object_name* Not Found

Explanation: This error occurs when rereading the DB2 catalog to get columns required for DDL generation.

User response: Call IBM Software Support if unable to resolve this error.

CKZ00136I Subtask *subtask_name*, DDL Create, Object Exists, Processing Continues, DDL *statement*

Explanation: This object already exists on the target.

User response: No action is required.

CKZ00137E No tt found in cache for index ccc.nnn

Explanation: The required object was not found in the cache. tt = object type ccc = object creator nnn = object name

User response: This error is probably due to the cache not being populated with the objects required. Check the database list for the source and target.

CKZ00138E Partition caching unable to find index ccc.nnn in the cache

Explanation: The required object was not found in the cache. ccc = object creator nnn = object name

User response: This error is probably due to the cache not being populated with the objects required. Check the database list for the source and target.

CKZ00139W SUBTASK-TERMINATION-WAIT Parm has an Invalid Value; SET Command Ignored

Explanation: This is a user error. The default will be used.

User response: Correct the error and resubmit the job.

CKZ00140I SUBTASK-TERMINATION-WAIT Parm Not in SET Command, Defaulting to ddd Seconds

Explanation: This indicates the default value will be used. ddd = decimal number of seconds to wait.

User response: No action is required.

CKZ00143I Subtask *subtask_number*, Log Apply Page Changes = *decimal_number_of_pages*

Explanation: Page with log records applied.

User response: No action is required.

CKZ00144W Subtask nn, Caching Requested for One or More Index Spaces Without a Preceding Table Space Type Specification in LISTDEF. Some or All Source or Target Objects May Not be Found.

Explanation: This message is a warning. It may be followed by CKZ54453E No Objects Selected. This message may or may not be printed once for each

subtask running. nn = subtask number

User response: Table space(s) must be specified in LISTDEF to use caching. Delete all index space LISTDEF statements and add ALWAYS-COPY-INDEXSPACES(Y) to the source job COPY command to allow DB2 Cloning Tool Table Space Cloning to create the index space LISTDEF commands internally.

CKZ00145W Subtask *subtask_number*, Inconsistent Bit Found in Page, DSN=*data_set_name*, Page=*DB2_page_number_in_hex*

Explanation: This message is a warning. It may or may not indicate a problem with page data.

User response: Determine if the page data is correct. If not, rerun the source and target job to reprocess the data set.

CKZ00151I HASH SI UTS ddd Table OBID Change - sss, ttt

Explanation: This message is informational. ddd = segment information UTS entry number sss = source table OBID in hex fff = target table OBID in hex

User response: No action is required.

CKZ00152I HASH Bucket ddd Table OBID Change - sss, ttt

Explanation: This message is informational. ddd = hash bucket entry number sss = source table OBID in hex fff = target table OBID in hex

User response: No action is required.

CKZ00153I HASH Collision ddd Table OBID Change - sss, ttt

Explanation: This message is informational. ddd = hash collision entry number sss = source table OBID in hex fff = target table OBID in hex

User response: No action is required.

CKZ00154I Tablespace prefetch skipped, no source DB2 list

Explanation: This message is informational.

User response: To prefetch table spaces, add SOURCE-PREFETCH-DATABASE-LIST to the CATALOG-PREFETCH COPY command in the source job.

CKZ00155E Tablespace ddd.sss not found in the cache

Explanation: The required table space object was not found in the cache. ddd = database name sss = table space name

User response: This error is probably due to the cache not being populated with the objects required. Check the database list for the source and target. Call IBM Software Support if unable to resolve this error.

CKZ00156E Database ddd not found in jjj prefetch list

Explanation: The required database object was not found in the cache. ddd = database name jjj = source or target

User response: Add the database to the source and/or target prefetch list.

CKZ00157I Tablespace prefetch skipped, no target DB2 list

Explanation: This message is informational.

User response: To prefetch table spaces, add TARGET-PREFETCH-DATABASE-LIST to the CATALOG-PREFETCH COPY command in the source job.

| **CKZ00158I Subtask *subtask_number*, cache disabled in TCP/IP server per DDL usage.**

| **Explanation:** This message is informational. Once DDL is scheduled to be output by the server job, target caching is disabled.

| **User response:** If you want to use target caching, restart the TCP/IP server.

| **CKZ00161I Subtask *subtask_number*, system page found, first, PAGE=*page_number*, DSN=*dsname*.**

| **Explanation:** A system page was found, but it was not the first page.

| **User response:** No action is required.

| **CKZ00166I Subtask *subtask_number*, RPL Error Msg: *message_text***

| **Explanation:** A message is printed from a VSAM macro return code other than zero.

| **User response:** No action is required.

**CKZ01000E ???????? SUPPORT MODULE MISSING
| ???????? DD STATEMENT MISSING**

Explanation: During program start, either required modules or DD statements were not discovered.

User response: For missing modules, ensure the proper //STEPLIBs are available. For missing DDs, add the appropriate DD statement to the execution JCL.

**CKZ01002E UNRECOGNIZABLE COMMAND (N)
ccccccccc**

Explanation: During command interpretation, command cccccccc could not be identified. As this situation can be determined in several places, sequence "n" isolates the point the error was detected.

User response: Check the spelling of the command.

CKZ01003E COMMAND HAS NO OPERANDS

Explanation: During command interpretation, the command preceding this message did not have any operands.

User response: Most commands have operands. Ensure continuation indicators are present if the command was continued onto a second line.

CKZ01005E FATAL ERRORS HAVE OCCURRED DURING //CKZINI PROCESSING.

Explanation: During program start, one or more problems occurred during decoding of the //CKZINI member. As the //CKZINI provides vital information for DB2 Cloning Tool, the program cannot continue execution.

User response: Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

CKZ01006E DB2 CLONING TOOL MUST EXECUTE AS A Z/OS APF AUTHORIZED PROGRAM

Explanation: During program start, DB2 Cloning Tool determined it was not z/OS APF authorized. Due to restraints set forth by z/OS, numerous functions will not work, thus the program cannot continue execution.

User response: Ensure that the library from which DB2 Cloning Tool is executing is z/OS APF authorized.

CKZ01007E INI AND MODULE RELEASES DO NOT MATCH.

Explanation: During program start, DB2 Cloning Tool determined that the release in the CKZINI member does not match the internal release. Processing terminates.

User response: Ensure that INIMERGE has been run.

CKZ01009I DB2 CLONING TOOL EXECUTION COMPLETE. HIGHEST RETURN CODE WAS n.

Explanation: Program termination message. "n" is the highest return encountered during processing of the commands. See other messages to determine the cause of a non-zero return code.

User response: No action is required.

CKZ01010E UNABLE TO DETERMINE MASTER CATALOG

Explanation: During program start, a problem occurred attempting to determine the name of the master catalog. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

CKZ01011E SWAREQ FAILED; R15=nnnnnnnn BLOCK REQUESTED=block

Explanation: An SWAREQ has failed for the indicated block. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ01020I PROGRAM: name info

Explanation: Displays the maintenance level of a loaded DB2 Cloning Tool program.

User response: No action is required.

CKZ01099E ABEND DURING DB2 CLONING TOOL PROCESSING

Explanation: During execution of a command within DB2 Cloning Tool, an abend took place that the command did not rectify.

User response: Ensure that all parameters on the named command are coded correctly, that external data sets used in the command are valid and try again. If the problem persists, contact IBM Software Support. Ensure all software run information is available: this includes the original JCL and control statements used to invoke DB2 Cloning Tool, and all the spooled output from its execution.

CKZ02001I hh:mm:ss COPY PROCESS COMPLETED; RETURN CODE=nnn

Explanation: COPY command processing message. For the 'completed' message, if the return code is non-zero, check other messages for errors and/or warnings.

User response: No action is required.

CKZ02001I hh:mm:ss COPY PROCESS COMPLETED; RETURN CODE=nnn

Explanation: COPY command processing message. For the 'completed' message, if the return code is non-zero, check other messages for errors and/or warnings.

User response: No action is required.

CKZ02003I DDNAME=ddname ALLOCATED FOR DSN=datasetname

Explanation: 'ddname' has been dynamically allocated for the specified data set.

User response: No action is required.

CKZ02004E DDNAME MISSING: ddname

Explanation: 'ddname' was specified for DB2 Cloning Tool to use. Processing terminates.

User response: Either correct the ddname specified, or, add the appropriate ddname to the job's JCL.

CKZ02005E ALLOCATION FAILED FOR DSN: datasetname

Explanation: Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ02005W DEALLOCATION FAILED FOR DDNAME: ddname

Explanation: Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ02006E ERROR CALLING CKZ01VV1 tttttt FUNCTION: function R15=nnnn R0=nnnnnnnn LOC=lllll

Explanation: A problem occurred using a dataspace. tttttt is the name of the internal table. lllll is the location where the error occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

CKZ02007W ERROR CALLING CKZ01HEX; FUNCTION: function R15=nnnn

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

CKZ02008E UNABLE TO LOAD PROGRAM:
program name

Explanation: The indicated program name was not found. Processing terminates.

User response: Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

CKZ02009E ERROR ACCESSING JOURNAL FILE;
LOC=IIIIII

Explanation: A VSAM error occurred accessing the journal file. Processing terminates.

User response: See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ02010E DUPLICATE JOURNAL ENTRY;
LOC=IIIIII

Explanation: A duplicate record was detected. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ02011E JOURNAL CONTROL RECORD NOT FOUND

Explanation: An expected record was not found in the DB2 Cloning Tool journal file. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ02017E THE DDNAME IS EMPTY OR HAS BEEN DUMMIED, DDNAME: ddn

Explanation: No records were read from the ddname specified for a keyword. Processing terminates.

User response: Check that the DD has not been specified as 'DD DUMMY' or 'DD DSN=NULLFILE'. Check that the DSN specified in the ddn has been created successfully.

CKZ02018E THE DDNAME INPUT HAS EXCEEDED THE CURRENT CAPACITY, DDNAME: ddn

Explanation: The number of entries read from the ddname exceeded the current capacity. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ02019E THE keyword DOES NOT HAVE A LRECL OF 80, DDNAME: ddn

Explanation: The data set allocated to the ddname for the keyword does not have a LRECL of 80. The LRECL of this data set must be 80.

User response: Change the data set allocated to the ddname to have a LRECL of 80.

CKZ02020I TARGET VOLUMES WILL BE CLIPPED WHEN THE VOLOPTIONS OFFLINECLIP COMMAND IS EXECUTED

Explanation: The target volumes will not be clipped by the COPY command due to the use of the VOLPAIRSDEVN-NOCLIP or VOLPAIRSDEVN-NOCLIP-DDN keywords. The target volumes will be clipped when the VOLOPTIONS OFFLINECLIP command is executed.

User response: No action is required.

CKZ02021I SOURCE USER CATALOGS WILL BE BACKED UP WHEN THE UCATOPTIONS BACKUP COMMAND IS EXECUTED

Explanation: The source user catalogs will not be backed by the COPY command due to the use of the USERCATALOGS-NOBACKUP keyword. The source user catalogs will be backed up when the UCATOPTIONS BACKUP command is executed.

User response: No action is required.

CKZ02022I VOLSER: volser LOGICAL NUMBER OF CYLINDERS: nnnnnnnn IS LESS THAN PHYSICAL NUMBER OF CYLINDERS: nnnnnnnn

Explanation: The identified volume has a logical size, from the Format 4 DSCB, that is less than the physical size, from the DCE. The logical size will be used for pairing this volume. This may lead to a condition where there are not enough target volumes available of the correct size to pair with all the source volumes.

User response: No action is required. ICKDSF can be used to make the logical size equal to the physical size.

| **CKZ2029I Log apply process was unable to connect to the server**

| **Explanation:** The log-apply process failed to connect to the server to output for output.

| **User response:** Ensure that the DB2 Cloning Tool server is active.

CKZ02030I DSS LEVEL=X'nnnnnnnn'

Explanation: The level of DSS returned by ADRMCLVL.

User response: No action is required.

CKZ02040E INVALID VALUE FOUND FOR item IN keyword RECORD: value

Explanation: An invalid value has been found for an item in a record in the data set allocated to the ddname for the keyword. The record is printed. Processing terminates.

User response: Correct the value for the item in the record to have a valid value.

CKZ02041E CKZ00900 UNEXPECTED RESULTS; error text

Explanation: An unexpected condition occurred calling program CKZ00900. 'error text' has a description of the problem. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ02043I ANTRQST LEVEL=nn; ESSRVCS LEVEL=nnn

Explanation: The level returned by ANTRQST REQUEST=LEVEL.

User response: No action is required.

CKZ02044W ANTRQST LEVEL NOT SUPPORTED; LEVEL=nn

Explanation: For FlashCopy support, the level must be greater than four (4). For SnapShot support, the level must be greater than one (1). The level returned by ANTRQST is not supported. Processing terminates.

User response: Check with your system programmer for upgrading the system.

CKZ02045E ANTMMAIN NOT ACTIVE

Explanation: An ANTRQST request failed because the system task, ANTMMAIN, is not active. Processing terminates.

User response: Start the system task, ANTMMAIN.

CKZ02046E ANTRQST LEVEL DOES NOT SUPPORT *function*

Explanation: The COPY command has keywords or parameters specified that request use of the identified function but the ANTRQST level does not support that function. Processing terminates.

User response: Check with your system programmer for upgrading the system.

CKZ02047E ANTRQST ERROR; request type RETURN CODE=nnnn X'hhhh' REASON CODE=nnnn X'hhhh' LOC=lllll VOLSER=volser

Explanation: An ANTRQST request failed. 'request type' indicates if the macro was incorrect, or, if the Data Mover failed the request. The return code and reason code are displayed in decimal and hexadecimal format. 'lllll' is an internal indicator of where the problem occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ02048I OPTIONS IN EFFECT FOR THIS EXECUTION: merged options

Explanation: Informational message indicating how COPY will handle the options. The displayed options are derived from the CKZINI and any overriding specifications in the command input.

User response: No action is required.

CKZ02049W keyword SPECIFIED BUT WILL NOT BE USED DUE TO reason

Explanation: The indicated keyword was specified but will not be used due to the indicated reason. Processing continues.

User response: Correct the keyword specifications.

CKZ02050E ERROR IN PARAMETERS FOR keyword

Explanation: The parameters for the indicated keyword were incorrect. Processing terminates.

User response: Check the keyword parameters. Mutually exclusive keywords may have been used.

CKZ02051E REQUIRED KEYWORD MISSING: keyword

Explanation: A keyword required for processing has been omitted. Processing terminates.

User response: Specify the required keyword.

CKZ02052E REQUIRED INI SECTION/TOKEN MISSING: SECTION=section TOKEN=token | REQUIRED INI VALUE MISSING FOR SECTION=section TOKEN=token | INVALID INI VALUE FOR SECTION=section TOKEN=token

Explanation: An error occurred validating the CKZINI

CKZ02053E • CKZ02062E

member options. Processing terminates.

User response: Correct the CKZINI member.

CKZ02053E **KEYWORD: keyword** **MAXIMUM LENGTH: nnn EXCEEDED | TOKEN: token** **MAXIMUM LENGTH: nnn EXCEEDED**

Explanation: The operand entered for a keyword or CKZINI token exceeded the maximum length allowed. nnn is the maximum allowed length. Processing terminates.

User response: Correct the length of the keyword's operand or the token's value.

CKZ02054E **KEYWORD HAS MORE THAN n OPERANDS; ONLY n ALLOWED: keyword**

Explanation: Multiple operands were detected for a keyword; only the indicated number of operands is permitted. Processing terminates.

User response: Correct the keyword to use the correct number of operands.

CKZ02055E **STORAGE GROUPS NOT SUPPORTED WITH DATA-MOVER PROGRAM NONE**

Explanation: The DATA-MOVER program was specified as 'NONE'. Storage group names/masks were specified for the source and/or target volume serials. Processing terminates.

User response: Correct the DATA-MOVER program specified, or, use keywords FROM-VOLSER/TO-VOLSER for the volume serials.

CKZ02056E **NOTHING SPECIFIED FOR KEYWORD: keyword**

Explanation: A keyword was entered without an appropriate operand. Processing terminates.

User response: Specify an appropriate operand for the keyword.

CKZ02057E **DUPLICATE FOUND; KEYWORD: keyword** **ENTRY: entry**

Explanation: The indicated 'entry' for the keyword was previously specified. Processing terminates.

User response: Remove the duplicate entry.

CKZ02058E **INVALID VALUE IN KEYWORD: keyword** **VALUE: value** **error text**

Explanation: The value in the keyword is invalid. 'error text' indicates the problem detected with the value. Processing terminates.

User response: Correct the value specified in the keyword.

CKZ02059E **KEYWORD: keyword1** **CANNOT BE USED WITH keyword2**

Explanation: The first indicated keyword cannot be used with the second indicated keyword. Processing terminates.

User response: Either remove the first indicated keyword or change the second indicated keyword to be compatible with the first indicated keyword.

CKZ02060E **UCBSCAN ERROR; RETURN CODE=nn** **REASON CODE=nn | UCBINFO ERROR; RETURN CODE=nn** **REASON CODE=nn**

Explanation: An error occurred using UCBSCAN or UCBINFO. Processing terminates.

User response: Contact IBM Software Support. Have available the listing containing this message.

CKZ02061E **CKZ01SMF ERROR; RETURN CODE=nnnn** **LOC: llllll** **entry**

Explanation: An error occurred using CKZ01SMF to obtain SSI information for the 'entry'. llllll is the internal location where the error occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing containing this message.

CKZ02062E *multiple possible messages; see Explanation*

Explanation:

NO STORAGE GROUPS RETURNED BY SSI

The COPY command was requested to determine volume serial numbers by storage group names. SSI did not return any storage groups to CKZ01SMF. Processing terminates.

NO VOLUME SERIALS RETURNED BY SSI

The COPY command was requested to check the SMS status of volume serials. SSI did not return any volume serials to CKZ01SMF. Processing terminates.

User response: Check that storage groups are defined on the system. If unable to determine the cause of this message, contact IBM Software Support. Have available the listing containing this message.

CKZ02063E EXPLICIT SOURCE STORAGE GROUP NOT FOUND | EXPLICIT VOLSER NOT FOUND ONLINE | MASK FOR SOURCE NOT RESOLVED

Explanation: The indicated 'entry' for the keyword was not matched. For VOL, the indicated volser, or, the volser derived from a storage group, was not found. For STG, the indicated storage group was not found.

User response: For an error with a source keyword, correct the keyword specification, or, ensure that all source volumes are online.

CKZ02063W NO VOL/STG MATCH FOUND; KEYWORD: keyword ENTRY: entry

Explanation: The indicated 'entry' for the keyword was not matched. For VOL, the indicated volser, or, the volser derived from a storage group, was not found. For STG, the indicated storage group was not found. Processing continues.

If the keyword involved source volumes, one of the CKZ02063E messages will be issued.

User response: No action is required.

CKZ02064E INVALID VOLSER: volser IN KEYWORD: keyword

Explanation: The volume serial number specified is invalid. Processing terminates.

User response: Correct the volser specification.

CKZ02065E STORAGE GROUP REFERENCED AS BOTH SOURCE AND TARGET: storagegroup name

Explanation: The storage group was used in both the FROM-STORAGEGROUP and TO-STORAGEGROUP keywords. Processing terminates.

User response: Correct the storage group specification.

CKZ02066E VOLSER REFERENCED AS BOTH SOURCE AND TARGET: volser

Explanation: The volume serial was used in both the FROM-VOLSER and TO-VOLSER keywords. Processing terminates.

User response: Correct the volume serial specification.

CKZ02067W NO VOLUME SERIALS FOR STORAGE GROUP: storage group

Explanation: SSI did not return any volsers for the storage group to CKZ01SMF. Processing continues.

User response: No action is required.

CKZ02068E UNMATCHED ENTRIES IN KEYWORD: keyword

Explanation: For USERCATALOGS, there must be a source BCS followed by a target BCS. An uneven number of BCS's was specified. For VOLPAIRS, there must be a source volume serial, target volume serial. An uneven number of entries was specified. For VOLPAIRSDEVN, there must be a source volume serial, target volume serial, target device number. Unmatched entries were found. Processing terminates.

User response: Correct the keyword specification.

CKZ02069E UNRESOLVED SYMBOL IN KEYWORD: keyword R15=nnnn ERR=error text

Explanation: A symbol was used in the keyword. CKZ01KSS was unable to resolve the symbol. Processing terminates.

User response: Define the symbol or remove the symbol.

CKZ02070E SECURITY PRODUCT DENIED ACCESS TO DSN: datasetname

Explanation: The dsn indicated is not authorized for alter by your security product. If the RACF profile that is associated is returned, it will be displayed. Processing terminates.

User response: Change the dsn to one you can use, or, have your security administrator give you 'ALTER' authority to the data set.

CKZ02071E RACROUTE ERROR; SAF RC=nnnn RACF RC=nnnn RACF REASON CODE=nnnn

Explanation: An unexpected return code from SAF or RACF occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing containing this message.

CKZ02072E MORE SOURCE VOLSER THAN TARGET VOLSER

Explanation: More volsers were found in the FROM-VOLSER or FROM-STORAGEGROUP keywords than were found in the TO-VOLSER or TO-STORAGEGROUP keywords. Processing terminates.

User response: Correct the FROM- or TO- specifications. There must be at least as many target volumes as source volumes.

CKZ02073E DATA-MOVER PGM NOT RECOGNIZED; PROGRAM=program

Explanation: A program was specified in the DATA-MOVER(PGM(..)) keyword. The program is not recognized. Processing terminates.

User response: Correct the program specification.

CKZ02074E USERCATALOG USED AS BOTH SOURCE AND TARGET: BCS dsname

Explanation: The BCS dsname was designated as a source in one pair and as a target in another pair. Processing terminates.

User response: Correct the USERCATALOGS specification.

CKZ02075E NO VOLUME SERIALS SELECTED FOR source/target

Explanation: No volume serials have been found for DB2 Cloning Tool to use as source volser, or, no volume serials have been found for DB2 Cloning Tool to use as target volser. Processing terminates.

User response: For 'SOURCE', check if volsers specified in the FROM- keyword have been removed by use of the EXCLUDE-FROM- keyword. For 'TARGET', check if volsers specified in the TO- keyword have been removed by use of the EXCLUDE-TO- keyword.

CKZ02076E DEVICE TYPE AND MODEL NOT FOUND FOR volser

Explanation: IOSCDs for the volser did not return information needed to pair source with target volsers.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ02077E EXCLUDE NOT ACCEPTED WITH DATA-MOVER PROGRAM NONE

Explanation: EXCLUDE-FROM-VOLSER and EXCLUDE-TO-VOLSER are not accepted with DATA-MOVER program 'NONE'. Processing terminates.

User response: Remove the EXCLUDE keyword.

CKZ02078E VOLPAIRSDEVN ONLY ACCEPTED WITH DATA-MOVER PROGRAM NONE or PPRC.

Explanation: Keyword VOLPAIRS was used, but, the DATA-MOVER program is not 'NONE'. Keyword VOLPAIRSDEVN was used, but, the DATA-MOVER program was neither 'NONE' nor 'PPRC'. Processing terminates.

User response: Correct the keyword specifications.

CKZ02078E VOLPAIRSDEVN ONLY ACCEPTED WITH DATA-MOVER PROGRAM NONE or PPRC.

Explanation: Keyword VOLPAIRS was used, but, the DATA-MOVER program is not 'NONE'. Keyword VOLPAIRSDEVN was used, but, the DATA-MOVER program was neither 'NONE' nor 'PPRC'. Processing terminates.

User response: Correct the keyword specifications.

CKZ02079E TARGET DEVICE NUMBER nnnn IS CURRENTLY ONLINE; VOLSER: volser

Explanation: Keyword VOLPAIRSDEVN was used, but, a specified target device number is online. Processing terminates.

User response: Check the VOLPAIRSDEVN specification. The target volume serials should be offline to the executing system and have the paired source volume serial in the label.

CKZ02080E TARGET DEVICE NUMBER nnnn PREVIOUSLY SELECTED FOR VOLSER: previous volser CURRENT VOLSER: current volser

Explanation: The same device number was used for more than one target volume serial. Processing terminates.

User response: Correct the VOLPAIRSDEVN specification.

CKZ02081E VOLUME volser DEVICE NUMBER nnnn HAS NO chpid PATHS AVAILABLE

Explanation: The indicated device has no paths available for DB2 Cloning Tool to use to access the volume. Processing terminates.

User response: Ensure that at least one chpid is online for each target device specified.

CKZ02082E TARGET DEVICE NUMBER: nnnn IS NOT AN ECKD DEVICE

Explanation: The indicated device does not appear to support ECKD™ commands. Processing terminates.

User response: Ensure that only supported devices are specified.

CKZ02083E FASTREP PARAMETER NOT RECOGNIZED; PARM=parameter

Explanation: The parameter specified in the DATA-MOVER(FASTREP(...)) keyword is not recognized. Processing terminates.

User response: Correct the parameter specification.

CKZ02084E OFFLINE SOURCES NOT ACCEPTED WITH DATA-MOVER PROGRAM pgm

Explanation: SOURCESOFFLINE(Y) was specified with DATA-MOVER(PGM(ADRDSU)) specified (or defaulted) or DATA-MOVER(PGM(EMCSNAP)) specified. The source volumes must be online for ADRDSU and EMCSNAP. Processing terminates.

User response: Either remove the SOURCESONLINE keyword, or specify DATA-MOVER(PGM(NONE)).

CKZ02085I DSNS FOR KEYWORD: keyword list of dsns

Explanation: Parsing found the listed dsns for the keyword.

User response: No action is required.

CKZ02086I STORAGE GROUPS/MASKS FOR KEYWORD: keyword

Explanation: Parsing found the listed storage groups/masks for the keyword.

User response: No action is required.

CKZ02087I nnnnn VOLUMES OR MASKS FOR KEYWORD: keyword nnnnn VOLUMES DERIVED FOR KEYWORD: keyword nnnnn VOLUMES RESOLVED FOR KEYWORD: keyword nnnnn VOLUME PAIRS FOR KEYWORD: keyword nnnnn VOLUMES/DEVICES FOR KEYWORD: keyword list of volsers

Explanation: Parsing found the listed volsers/masks for a keyword, or, derived the listed volsers from a storage group keyword. nnnnn is the number of volume serials. The third format, RESOLVED, indicates the number of volume serials found online for the associated keyword.

User response: No action is required.

CKZ02088I VOLUME SERIALS TO BE USED FOR SOURCE | VOLUME SERIALS TO BE USED FOR TARGET list of volsers

Explanation: The listed volsers will be used during DB2 Cloning Tool COPY processing.

User response: No action is required.

CKZ02089I TARGET VOLUMES WILL NOT BE CHECKED FOR EMPTY

Explanation: Informational message.

User response: No action is required.

CKZ02090I TARGET VOLUMES WILL BE CHECKED FOR EMPTY

Explanation: Informational message.

User response: No action is required.

CKZ02091I VALIDATING KEYWORD: keyword

Explanation: Parsing is checking the indicated keyword indicated in the command.

User response: No action is required.

CKZ02101I hh:mm:ss VOLUME COPY STARTED - PROGRAM REV=rrr (SIMULATION **) | hh:mm:ss VOLUME COPY COMPLETED; RETURN CODE=nnn**

Explanation: VOLUME PAIRING processing message.

User response: No action is required.

CKZ02104E OPEN FAILED FOR DDNAME=ddname

Explanation: 'ddname' was allocated for DB2 Cloning Tool to use, but, the open for the file failed. Processing terminates.

User response: If unable to determine the reason the open failed, contact IBM Software Support. Have available the listing that contains this message.

CKZ02107W ERROR CALLING CKZ01HEX; FUNCTION: function R15=nnnn

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

CKZ02108E UNABLE TO LOAD PROGRAM: program name

Explanation: The indicated program name was not found. Processing terminates.

User response: Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

**CKZ02109E ERROR ACCESSING JOURNAL FILE;
LOC=IIIII**

Explanation: A VSAM error occurred accessing the journal file. Processing terminates.

User response: See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ02110E DUPLICATE JOURNAL ENTRY;
LOC=IIIII**

Explanation: A duplicate record was detected. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ02111E JOURNAL VOLUME PAIR RECORD(S)
NOT FOUND | JOURNAL UCAT PAIR
RECORD(S) NOT FOUND**

Explanation: An expected record was not found in the DB2 Cloning Tool journal file. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ02112E JOURNAL UCAT PAIR RECORD IS
WRONG VERSION**

Explanation: The journal record does not match the expected format. The record is printed. Processing terminates.

User response: Verify that different releases of DB2 Cloning Tool have not been run using the same journal data set. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains this message.

**CKZ02130I ADRDSSU TASK COMPLETED;
RETURN CODE=nn SYSOUT
DD=ddname**

Explanation: An ADRDSSU task has ended.

User response: No action is required if the RETURN CODE is zero. If the RETURN CODE is not zero, check the indicated sysout file for warning or error messages for the volume copy task.

**CKZ02131I PRE-COPIED VOLUME PAIRS
ACCEPTED: SOURCE TARGET**

Explanation: The DATA-MOVER program was NONE. The source and target volume serials have been paired as entered.

User response: No action is required.

**CKZ02138I UNEXPECTED RESULTS FROM
FCQUERY FQMAP; VOLSER= volser**

Explanation: An ANTRQST FCQUERY FQMAP call returned unexpected results.

User response: Please report this message to IBM Software Support.

CKZ02140E UNEXPECTED CONDITION; error text

Explanation: An unexpected condition occurred while dispatching a copy task or while waiting for the completion of a copy task. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ02141E INTERNAL ERROR; DATA MOVER
SETTING xx INVALID**

Explanation: The setting for the data mover to be used was not recognized. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ02142E MORE SOURCE VOLSERS THAN
TARGET VOLSERS**

Explanation: Volume pairing detected more source volumes than target volumes. Processing terminates.

User response: This may have been caused by the use of TARGET-VOLS-SHOULD-BE-EMPTY eliminating target volumes. If unable to determine the cause, contact IBM Software Support. Have available the listing that contains this message.

CKZ02144E ANTRQST LEVEL NOT SUPPORTED

Explanation: For FlashCopy support, the level must be greater than four (4). For SnapShot support, the level must be greater than one (1). The level returned by ANTRQST is not supported. Processing terminates.

User response: Check with your system programmer for upgrading the system.

CKZ02145E ANTMMAIN NOT ACTIVE

Explanation: An ANTRQST request failed because the system task, ANTMMAIN, is not active. Processing terminates.

User response: Start the system task, ANTMMAIN.

**CKZ02146E ANTRQST DID NOT RETURN ANY
INFORMATION; RESTART ANTMMAIN
SYSTEM TASK**

Explanation: The ANTRQST REQUEST=FCQUERY

did not receive information for a device. Processing terminates.

User response: Restart the ANTRQST system task, or, apply IBM APAR OW47323.

**CKZ02147E ANTRQST ERROR; request type
RETURN CODE=nnnn X'hhhh'
REASON CODE=nnnn X'hhhh'
LOC=lllll VOLSER=volser**

Explanation: An ANTRQST request failed. 'request type' indicates if the macro was incorrect, or, if the Data Mover failed the request. The return code and reason code are displayed in decimal and hexadecimal format. 'lllll' is an internal indicator of where the problem occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ02148E SOURCE VOLUME NOT ELIGIBLE
FOR PROCESSING; VOLSER=volser**

Explanation: A request for SDVCINFO or SQRVDVCS indicated the indicated volume serial is not eligible for FlashCopy or SnapShot. 'reason text' indicates the system response. For a source volume, processing terminates.

User response: Eliminate source volumes which are not eligible for FlashCopy or SnapShot.

**CKZ02148I TARGET VOLUME NOT ELIGIBLE
FOR PROCESSING; VOLSER=volser
WILL BE SKIPPED. 'reason text'**

Explanation: A request for SDVCINFO or SQRVDVCS indicated the indicated volume serial is not eligible for FlashCopy or SnapShot. 'reason text' indicates the system response. The target volume is eliminated for pairing; processing continues.

User response: No action is required.

**CKZ02149E SOURCE VOLSER: volser ACTIVE IN
COPY RELATIONSHIP**

Explanation: An FCQUERY request indicated the volser is currently in a copy relationship. The FCQUERY response is printed. For a source volser, processing terminates.

The format of the FCQUERY response for FlashCopy V1 is: devn,ssid,lss,cca,cu,serial,status

Where:

- devn is the device number
- ssid is the subsystem ID for the device
- lss is logical subsystem number
- cca is the subsystem device address
- cu is the subsystem type number

- serial is the subsystem serial number
- status is the current status of the device:

XRC Device is in XRC pair

PPRC Device is in PPRC pair

CC Device is in CC session

FC Device is in FlashCopy (no background copy)

FC..xxx%
Device is in FlashCopy (background copy)

SIMPLEX
Device is not in any copy status

The format of the FCQUERY response for FlashCopy V2 is: devn,ssid,lss,cca,cu,serial,act,max,xc,pc,cc

Where:

- devn is the device number
- ssid is the subsystem ID for the device
- lss is logical subsystem number
- cca is the subsystem device address
- cu is the subsystem type number
- serial is the subsystem serial number
- act is the current number of FlashCopy relationships that the device has. This value is indicated in decimal format.
- max is the maximum number of FlashCopy relationships that the device may have. This value is indicated in decimal format.
- xc is either S indicating that the device is an XRC source volume, or is N indicating that the device is not an XRC source volume
- pc is either P indicating that the device is a PPRC primary volume, is S indicating that the device is an PPRC secondary volume, or is N for neither
- cc is either S indicating that the device is a concurrent copy source volume, or is N indicating that the device is not a concurrent copy source volume

User response: Exclude that source volume from processing, or, ensure that all copy relationships have terminated before initiating the COPY.

**CKZ02149I TARGET VOLSER: volser ACTIVE IN
COPY RELATIONSHIP**

Explanation: An FCQUERY request indicated the volser is currently in a copy relationship. The FCQUERY response is printed. For a target volser, processing continues, but that target volser will not be used by DB2 Cloning Tool.

The format of the FCQUERY response for FlashCopy V1 is: devn,ssid,lss,cca,cu,serial,status

Where:

CKZ02150E • CKZ02154I

- devn is the device number
- ssid is the subsystem ID for the device
- lss is logical subsystem number
- cca is the subsystem device address
- cu is the subsystem type number
- serial is the subsystem serial number
- status is the current status of the device:

XRC Device is in XRC pair
PPRC Device is in PPRC pair
CC Device is in CC session
FC Device is in FlashCopy (no background copy)
FC..xxx%
Device is in FlashCopy (background copy)
SIMPLEX
Device is not in any copy status

The format of the FCQUERY response for FlashCopy V2 is: devn,ssid,lss,cca,cu,serial,act,max,xc,pc,cc

Where:

- devn is the device number
- ssid is the subsystem ID for the device
- lss is logical subsystem number
- cca is the subsystem device address
- cu is the subsystem type number
- serial is the subsystem serial number
- act is the current number of FlashCopy relationships that the device has. This value is indicated in decimal format.
- max is the maximum number of FlashCopy relationships that the device may have. This value is indicated in decimal format.
- xc is either S indicating that the device is an XRC source volume, or is N indicating that the device is not an XRC source volume
- pc is either P indicating that the device is a PPRC primary volume, is S indicating that the device is an PPRC secondary volume, or is N for neither
- cc is either S indicating that the device is a concurrent copy source volume, or is N indicating that the device is not a concurrent copy source volume

User response: Exclude that source volume from processing, or, ensure that all copy relationships have terminated before initiating the COPY.

CKZ02150E NO INFORMATION RETURNED FOR SOURCE VOLSER=volser

Explanation: SDVCINFO did not return information for the indicated source volume. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ02151E NO TARGETS RETURNED FOR SOURCE VOLSER=volser

Explanation: SQRVDVCS did not return any target volsers for the indicated source volume. Processing terminates.

User response: Ensure that for every source volume, there is a target volume that meets the criteria required for FlashCopy or Snapshot. If unable to determine the reason for this message, contact IBM Software Support. Have available the listing that contains this message.

CKZ02152E NO TARGET VOLSER FOUND FOR SOURCE VOLSER=volser source volser information returned volser(s) information

Explanation: Based on the source volume's device type, FlashCopy or SnapShot capability, LSS, and number of tracks, no suitable target volume was found for a COPY FULL.

User response: Ensure that for every source volume, there is a target volume that meets the criteria required. If volser masks were used, you may need to explicitly code volume serial pairs in the FROM/TO parameters to ensure particular volumes are paired. If unable to determine the reason for this message, contact IBM Software Support. Have available the listing that contains this message.

CKZ02153E ERROR DURING VVAS; RC=X'nnnn' REASON=X'nnnn' VOLSER=volser

Explanation: An error occurred using VVAS to check a target volume. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ02154I TARGET VOLUME ELIMINATED, VOLSER=volser | DATA SET OTHER THAN SYS1.VTOCIX OR SYS1.VVDS FOUND: datasetname

Explanation: An unexpected data set was found on a target volume in response to TARGET-VOLS-SHOULD-BE-EMPTY. Processing continues.

User response: No action is required, unless this causes fewer target volumes than source volumes for the pairing process.

CKZ02155I SMS SOURCE VOLSER=volser HAS BEEN PAIRED TO NON SMS TARGET VOLSER=volser | NON SMS SOURCE VOLSER=volser HAS BEEN PAIRED TO SMS TARGET VOLSER=volser

Explanation: A source volume has been paired to a target volume that does not have the same SMS attribute. There may be problems when accessing or deleting data sets on the target volume.

User response: No action is required.

CKZ02156E SOURCE VOLSER: volser NOT FLASHCOPY CAPABLE

Explanation: An FCQUERY request indicated the volser is not FlashCopy capable. Information returned by FCQUERY is printed. For a source volser, processing terminates.

User response: Exclude that source volume from processing, or, determine why the volume is not FlashCopy capable and correct if possible.

CKZ02156I TARGET VOLSER: volser NOT FLASHCOPY CAPABLE

Explanation: An FCQUERY request indicated the volser is not FlashCopy capable. Information returned by FCQUERY is printed. For a target volser, processing continues, but that target volser will not be used.

User response: Exclude that source volume from processing, or, determine why the volume is not FlashCopy capable and correct if possible.

CKZ02158I SLOW COPY WILL BE USED FROM SOURCE VOLSER= volser TO TARGET VOLSER= volser

Explanation: FASTREP(PREF) has been specified and the copy between the two indicated volumes will be a slow copy rather than a fast FlashCopy or Snapshot.

User response: No action is required.

CKZ02160E THE VOLSER SPECIFIED FOR A SOURCE USERCATALOG IS NOT A SOURCE VOLSER VOLSER=volser BCS=catalogname

Explanation: The volser specified for a source catalog in the USERCATALOGS keyword is not a source volser. Processing terminates.

User response: Correct the volser specified for the source catalog in the USERCATALOGS keyword to be source volser.

CKZ02161I SOURCE USERCATALOG WILL BE READ FROM TARGET VOLUME; VOLSER=volser BCS=catalogname

Explanation: The source catalog will be read from the shown target volume.

User response: No action is required.

CKZ02201I hh:mm:ss BCS BACKUP STARTED - PROGRAM REV=rrr | hh:mm:ss BCS BACKUP COMPLETED; RETURN CODE=nnn

Explanation: BCS backup processing message.

User response: No action is required.

CKZ02205E ALLOCATION FAILED FOR DSN: datasetname | DEALLOCATION FAILED FOR DSN: datasetname

Explanation: Dynamic allocation for a data set failed, or, dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing terminates.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ02206E ERROR CALLING CKZ01VV1 tttttt FUNCTION: function R15=nnnn R0=nnnnnnnn LOC=lllll

Explanation: A problem occurred using a dataspace. tttttt is the name of the internal table. lllll is the location where the error occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

CKZ02207W ERROR CALLING CKZ01HEX; FUNCTION: function R15=nnnn

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

CKZ02209E ERROR ACCESSING JOURNAL FILE; LOC=lllll

Explanation: A VSAM error occurred accessing the journal file. Processing terminates.

User response: See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM

Software Support. Have available the listing that contains these messages.

CKZ02211E JOURNAL CONTROL RECORD NOT FOUND | JOURNAL USER CATALOG RECORD(S) NOT FOUND

Explanation: An expected record was not found in the DB2 Cloning Tool journal file. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ02212E JOURNAL CONTROL RECORD IS WRONG VERSION | JOURNAL UCAT PAIR RECORD IS WRONG VERSION

Explanation: The journal record does not match the expected format. The record is printed. Processing terminates.

User response: Verify that different releases of DB2 Cloning Tool have not been run using the same journal data set. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ02213E COUNT MISMATCH; RECORDS READ=rrrr PARM COUNT=pppp

Explanation: The number of User Catalog records read from the DB2 Cloning Tool journal, rrrr, is not the same as the number indicated in the DB2 Cloning Tool journal control record, pppp. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ02230I BCS BACKUP TASK COMPLETED; RETURN CODE=nn SYSOUT DD=ddname

Explanation: A BCS backup task has ended.

User response: No action is required if the RETURN CODE is zero. If the RETURN CODE is not zero, check the indicated sysout file for warning or error messages for the BCS backup task.

CKZ02240E UNEXPECTED CONDITION; error text

Explanation: An unexpected condition occurred while dispatching a BCS backup task or while waiting for the completion of a BCS backup task. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ02241I BCS=bcs dsname WILL BE BACKED UP TO DSN=backup dsname

Explanation: The indicated BCS will be backed up to the indicated backup data set name. The backup data set name was derived from the CATWORK-DSN parameter.

User response: No action is required.

CKZ02242E ERROR PARSING CATWORK-DSN: catwork dsn

Explanation: An internal error occurred handling the CATWORK-DSN.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ02301I hh:mm:ss EMC VOLUME COPY STARTED - PROGRAM REV=rrr (SIMULATION **) | hh:mm:ss EMC VOLUME COPY COMPLETED; RETURN CODE=nnn**

Explanation: EMC VOLUME PAIRING processing message.

User response: No action is required.

CKZ02304E OPEN FAILED FOR DDNAME=ddname

Explanation: 'ddname' was allocated for DB2 Cloning Tool to use, but the open for the file failed. Processing terminates.

User response: If unable to determine the reason the open failed, contact IBM Software Support. Have available the listing that contains this message.

CKZ02307W ERROR CALLING CKZ01HEX; FUNCTION: function R15=nnnn

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

CKZ02308E UNABLE TO LOAD PROGRAM: program name

Explanation: The indicated program name was not found. Processing terminates.

User response: Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

**CKZ02309E ERROR ACCESSING JOURNAL FILE;
LOC=IIIII**

Explanation: A VSAM error occurred accessing the journal file. Processing terminates.

User response: See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ02310E DUPLICATE JOURNAL ENTRY;
LOC=IIIII**

Explanation: A duplicate record was detected. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains these messages.

**CKZ02311E JOURNAL VOLUME PAIR RECORD(S)
NOT FOUND | JOURNAL UCAT PAIR
RECORD(S) NOT FOUND**

Explanation: An expected record was not found in the DB2 Cloning Tool journal file. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains these messages.

**CKZ02312E JOURNAL UCAT PAIR RECORD IS
WRONG VERSION**

Explanation: The journal record does not match the expected format. The record is printed. Processing terminates.

User response: Verify that different releases of DB2 Cloning Tool have not been run using the same journal data set. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ02330I EMCSNAPI COMPLETED; RETURN
CODE=*return_code***

Explanation: An EMCSNAPI call ended.

User response: If the return code is 0, no action is required. If the return code is not 0, check the SYSOUT file for warning or error messages that are related to the volume snap. A return code of 4 may result if any target volume is online to another system.

**CKZ02340E VOLP ENTRY NOT FOUND FOR
VOLSER=*volser***

Explanation: An unexpected condition occurred while processing. A previously found VOLP entry cannot be found. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ02341E INTERNAL ERROR; DATA MOVER
SETTING *xx* INVALID**

Explanation: The setting for the data mover to be used was not recognized. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ02342E MORE SOURCE VOLSERS THAN
TARGET VOLSERS**

Explanation: Volume pairing detected more source volumes than target volumes. Processing terminates.

User response: This may have been caused by the use of TARGET-VOLS-SHOULD-BE-EMPTY eliminating target volumes or by some of the target volumes not having the necessary capabilities. If unable to determine the cause, contact IBM Software Support. Have available the listing that contains this message.

CKZ02343I *module-name* VERSION *version*

Explanation: This message reports the version information retrieved from the indicated module.

User response: No action is required.

CKZ02345E EMCSCF NOT ACTIVE; LOC=IIIII

Explanation: An EMC request failed because the system task, EMCSCF, is not active. Processing terminates.

User response: Start the system task, EMCSCF.

**CKZ02346E *module-name* ERROR; request type
R15=*nnnn* X'*hhhh*' LOC=IIIII
VOLSER=*volser***

Explanation: The call to the indicated module for the indicated request type failed. The return code is displayed in decimal and hexadecimal format. 'IIIII' is an internal indicator of where the problem occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ02347E *module-name* ERROR; request type
RETURN CODE=*nnnn* X'*hhhh*'
REASON CODE=*nnnn* X'*hhhh*'
LOC=IIIII VOLSER=*volser***

Explanation: The call to the indicated module for the indicated request type failed. The return code and reason code are displayed in decimal and hexadecimal format. 'IIIII' is an internal indicator of where the problem occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ02348E SOURCE VOLUME NOT ELIGIBLE FOR PROCESSING; VOLSER=volser

Explanation: It has been determined that the indicated volume serial is not eligible for use. 'reason text' indicates the reason. Processing terminates.

User response: Eliminate source volumes which are not eligible for use.

CKZ02348I TARGET VOLUME NOT ELIGIBLE FOR PROCESSING; VOLSER=volser WILL BE SKIPPED. 'reason text'

Explanation: It has been determined that the indicated volume serial is not eligible for use. 'reason text' indicates the reason. Processing continues.

User response: No action is required.

CKZ02352E NO TARGET VOLSER FOUND FOR SOURCE VOLSER=volser source volser information target volser(s) information

Explanation: Based on the source volume's device type, EMC SNAP capability, Storage Subsystem, and number of tracks, no suitable target volume was found for a SNAP VOLUME.

User response: Ensure that for every source volume, there is a target volume that meets the criteria required. If volser masks were used, you may need to explicitly code volume serial pairs in the FROM/TO parameters to ensure particular volumes are paired. If unable to determine the reason for this message, contact IBM Software Support. Have available the listing that contains this message.

CKZ02353E ERROR DURING VVAS; RC=X'n'nnn' REASON=X'n'nnn' VOLSER=volser

Explanation: An error occurred using VVAS to check a target volume. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ02354I TARGET VOLUME ELIMINATED, VOLSER=volser | DATA SET OTHER THAN SYS1.VTOCIX OR SYS1.VVDS FOUND: datasetname

Explanation: An unexpected data set was found on a target volume in response to TARGET-VOLS-SHOULD-BE-EMPTY. Processing continues.

User response: No action is required, unless this causes fewer target volumes than source volumes for the pairing process.

CKZ02355I SMS SOURCE VOLSER=volser HAS BEEN PAIRED TO NON SMS TARGET VOLSER=volser | NON SMS SOURCE VOLSER=volser HAS BEEN PAIRED TO SMS TARGET VOLSER=volser

Explanation: A source volume has been paired to a target volume that does not have the same SMS attribute. There may be problems when accessing or deleting data sets on the target volume.

User response: No action is required.

CKZ02356I CURRENT SNAP RELATIONSHIPS FOR type VOLSER: volser volser relationship information

Explanation: The current snap relationships for the volser are listed. 'type' indicates if the volser is a SOURCE or TARGET volume.

User response: No action is required.

CKZ02360E THE VOLSER SPECIFIED FOR A SOURCE USERCATALOG IS NOT A SOURCE VOLSER VOLSER=volser BCS=catalogname

Explanation: The volser specified for a source catalog in the USERCATALOGS keyword is not a source volser. Processing terminates.

User response: Correct the volser specified for the source catalog in the USERCATALOGS keyword to be source volser.

CKZ02361I SOURCE USERCATALOG WILL BE READ FROM TARGET VOLUME; VOLSER=volser BCS=catalogname

Explanation: The source catalog will be read from the shown target volume.

User response: No action is required.

CKZ02501I hh:mm:ss COPYRESTART STARTED - PROGRAM REV=rrr | hh:mm:ss COPYRESTART COMPLETED; RETURN CODE=nnn

Explanation: COPYRESTART processing message.

User response: No action is required.

CKZ02503I DDNAME=ddname ALLOCATED FOR DSN=datasetname

Explanation: 'ddname' has been dynamically allocated for the indicated data set.

User response: No action is required.

CKZ02504E DDNAME MISSING: ddname

Explanation: 'ddname' was specified for DB2 Cloning Tool to use. Processing terminates.

User response: Either correct the ddname specified, or, add the appropriate ddname to the job's JCL.

**CKZ02505E ALLOCATION FAILED FOR DSN:
datasetname**

Explanation: Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ02505W DEALLOCATION FAILED FOR
DDNAME: ddname**

Explanation: Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ02506E ERROR CALLING CKZ01VV1 tttttt
FUNCTION: function R15=nnnn
R0=nnnnnnnn LOC=llll**

Explanation: A problem occurred using a dataspace. tttttt is the name of the internal table. llll is the location where the error occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

**CKZ02507W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn**

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

**CKZ02508E UNABLE TO LOAD PROGRAM:
program name**

Explanation: The indicated program name was not found. Processing terminates.

User response: Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

**CKZ02509E ERROR ACCESSING JOURNAL FILE;
LOC=lllll**

Explanation: A VSAM error occurred accessing the journal file. Processing terminates.

User response: See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ02511E JOURNAL CONTROL RECORD NOT
FOUND | JOURNAL VOLUME PAIR
RECORD(S) NOT FOUND**

Explanation: An expected record was not found in the DB2 Cloning Tool journal file. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ02513E RECORD COUNT IS ZERO; LOC=lllll
| COUNT MISMATCH; RECORDS
READ=rrrr CONTROL RECORD
COUNT=cccc**

Explanation: There was a problem with the journal records needed to initiate the command. For the first format, the journal control record indicates no entries were added. For the second format, the number of records read from the journal, rrrr, is not the same as the number indicated in the journal control record, cccc. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ02515E THE COPY PROCESS WAS NOT
INITIATED BY A METHOD WHICH IS
SUPPORTED BY RESTART**

Explanation: Only a copy initiated by PPRC where the pairs have not been terminated may be restarted. Processing terminates.

User response: If PPRC was used to initiated the volume copies and the copies are still in progress, contact IBM Software Support. Have available the listing that contains this message.

CKZ02516E COPY STATUS IS n

Explanation: Only a copy initiated by PPRC supports the COPYRESTART command. The journal indicates that PPRC was not initiated by DB2 Cloning Tool. Processing terminates.

User response: If PPRC was used as the COPY DATA-MOVER PGM, contact IBM Software Support. Have available the listing that contains this message.

CKZ02531W COPY STEP WAS A SIMULATION

Explanation: The journal indicates that the COPY command was a simulation. No restart of the copy will be done.

User response: No action is required.

**CKZ02551E REQUIRED KEYWORD MISSING:
keyword**

Explanation: A keyword required for processing has been omitted. Processing terminates.

User response: Specify the required keyword.

**CKZ02553E KEYWORD: keyword MAXIMUM
LENGTH: nnn EXCEEDED**

Explanation: The operand entered for a keyword exceeded the maximum length allowed for the operand. nnn is the maximum allowed length for the keyword. Processing terminates.

User response: Correct the length of the keyword's operand.

**CKZ02554E KEYWORD HAS MORE THAN 1
OPERAND; ONLY 1 ALLOWED:
keyword**

Explanation: Multiple operands were detected for a keyword; only one operand is permitted. Processing terminates.

User response: Correct the keyword to use one operand.

**CKZ02556E NOTHING SPECIFIED FOR
KEYWORD: keyword**

Explanation: A keyword was entered without an appropriate operand. Processing terminates.

User response: Specify an appropriate operand for the keyword.

**CKZ02557E DUPLICATE FOUND; KEYWORD:
keyword ENTRY: entry**

Explanation: The indicated 'entry' for the keyword was previously specified. Processing terminates.

User response: Remove the duplicate entry.

**CKZ02558E INVALID VALUE IN KEYWORD:
keyword VALUE: value error text**

Explanation: The value in the keyword is invalid. 'error text' indicates the problem detected with the value. Processing terminates.

User response: Correct the value specified in the keyword.

**CKZ02564E INVALID VOLSER: volsr IN
KEYWORD: keyword**

Explanation: The volume serial number indicated is invalid. Processing terminates.

User response: Correct the volsr specification.

**CKZ02568E UNPAIRED ENTRIES IN KEYWORD:
keyword**

Explanation: For REPL-TGT-DEVN, there must be a target volume serial followed by a target device number. An uneven number of entries was specified. Processing terminates.

User response: Correct the keyword specification.

**CKZ02587I nnnnn VOLUMES/DEVICES FOR
KEYWORD: keyword list of volsers**

Explanation: Parsing found the listed volsers/device numbers for a keyword. nnnnn is the number of entries.

User response: No action is required.

**CKZ02601I hh:mm:ss PPRC RESTART STARTED -
PROGRAM REV=rrr | hh:mm:ss PPRC
RESTART COMPLETED; RETURN
CODE=nnn**

Explanation: PPRC RESTART processing message.

User response: No action is required.

**CKZ02607W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn**

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

**CKZ02608E UNABLE TO LOAD PROGRAM:
program name**

Explanation: The indicated program name was not found. Processing terminates.

User response: Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

**CKZ02611E NO MATCH FOUND IN VOLUME
PAIRS FOR VOLSER targetvolsr**

Explanation: No matching target volume serial number was found in the originally specified VOLPAIRSDEVN with what was specified in REPL-TGT-DEVN. Processing terminates.

User response: Correct the specified target volume serial number.

CKZ02632I PPRC RELATIONSHIP ESTABLISHED;
SOURCE: sourcevolser
sourcedevicenumber TARGET:
targetvolser targetdevicenumber

Explanation: A new PPRC relationship has been established between the indicated volumes.

User response: No action is required.

CKZ02644E ANTRQST LEVEL NOT SUPPORTED

Explanation: For PPRC support, the level must be one (1). The level returned by ANTRQST is not supported. Processing terminates.

User response: Check with your system programmer for upgrading the system.

CKZ02645E system task NOT ACTIVE

Explanation: An ANTRQST request failed because a system task is not active. Processing terminates.

User response: Start the indicated system task.

CKZ02646E ANTRQST DID NOT RETURN ANY INFORMATION; RESTART system task SYSTEM TASK

Explanation: The ANTRQST REQUEST=PQUERY did not receive information for a device. Processing terminates.

User response: Restart the indicated system task.

CKZ02647E ANTRQST ERROR; request type RETURN CODE=nnnn X'hhhh' REASON CODE=nnnn X'hhhh' LOC=lllll

Explanation: An ANTRQST request failed. 'request type' indicates if the macro was incorrect, or, if the Data Mover failed the request. The return code and reason code are displayed in decimal and hexadecimal format. 'lllll' is an internal indicator of where the problem occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ02650E UNRECOGNIZED STATUS FROM PQUERY

Explanation: The device status returned by PQUERY was not recognized. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ02920W MESSAGE NUMBER NOT FOUND; nnnn x'nnnn'

Explanation: The message number for an ANTRQST error was not found in an internal messages table. Processing continues.

User response: Please report this message to IBM Software Support.

CKZ03001I hh:mm:ss COPY STARTED - PROGRAM REV=rrr | hh:mm:ss COPY COMPLETED; RETURN CODE=nnn

Explanation: COPY task processing message.

User response: No action is required.

CKZ03003I DDNAME=ddname ALLOCATED FOR DSN=datasetname

Explanation: 'ddname' has been dynamically allocated for the indicated data set.

User response: No action is required.

CKZ03004E OPEN FAILED FOR DDNAME=ddname

Explanation: 'ddname' was allocated for DB2 Cloning Tool to use, but, the open for the file failed. Processing terminates.

User response: If unable to determine the reason the open failed, contact IBM Software Support. Have available the listing that contains this message.

CKZ03005E ALLOCATION FAILED FOR DSN: datasetname | ALLOCATION FAILED FOR DDNAME: ddname

Explanation: Dynamic allocation for a data set or ddname failed. The associated z/OS messages are displayed. Processing terminates.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ03005W DEALLOCATION FAILED FOR DDNAME: ddname

Explanation: Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ03008E UNABLE TO LOAD PROGRAM:
program name

Explanation: The indicated program name was not found. Processing terminates.

User response: Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

CKZ03012E UNABLE TO ESTABLISH ESTAEX;
R15=nnnn

Explanation: The program was not able to establish an estaex environment. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ03021E ADRDSSU COPY FAILED; R15=nnnn

Explanation: A non-zero return code was received from ADRDSSU. The ADRDSSU messages will be contained in the sysout for the copy task. Processing terminates.

User response: Check the ADRDSSU messages for the cause of the failure. If assistance is required, contact IBM Software Support. Have available the listing that contains this message.

CKZ03022E DATA MOVER SETTING IS INVALID;
VALUE= value

Explanation: An internal error has occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ03023E INTERNAL ERROR; reason

Explanation: An internal error has occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ03041E CKZ00900 UNEXPECTED RESULTS;
error text

Explanation: An unexpected condition occurred calling program CKZ00900. 'error text' has a description of the problem. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ03042E NO MATCH FOUND FOR DSS TASK
NUMBER: nnn

Explanation: The DSS task number was not matched to any of the volume pairs for this task. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ03043E NO MATCH FOUND FOR TASK DD
IN VOLUME PAIRS

Explanation: The copy task was dispatched, but, no volume pairs had been assigned to it. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ03099E ABEND DURING COPY COMMAND

Explanation: An abend occurred for a copy task. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ03101I hh:mm:ss CLIP TARGETS STARTED -
PROGRAM REV=rrr (SIMULATION**
****) | hh:mm:ss CLIP TARGETS**
COMPLETED; RETURN CODE=nnn

Explanation: Offline target processing message.

User response: No action is required.

CKZ03103I DDNAME=ddname ALLOCATED FOR
DSN=datasetname

Explanation: 'ddname' has been dynamically allocated for the indicated data set.

User response: No action is required.

CKZ03104E OPEN FAILED FOR DDNAME=ddname

Explanation: 'ddname' was allocated for DB2 Cloning Tool to use, but, the open for the file failed. Processing terminates.

User response: If unable to determine the reason the open failed, contact IBM Software Support. Have available the listing that contains this message.

CKZ03105E ALLOCATION FAILED FOR DSN:
datasetname

Explanation: Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact

IBM Software Support. Have available the listing containing these messages.

**CKZ03105W DEALLOCATION FAILED FOR
DDNAME: ddname**

Explanation: Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ03108E UNABLE TO LOAD PROGRAM:
program name**

Explanation: The indicated program name was not found. Processing terminates.

User response: Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

**CKZ03131E CAPTURE FOR UCB FAILED; UCB
ADDRESS: nnnnnnnn R15: nnnn
TARGET VOLUME SERIAL: volser**

Explanation: An attempt to use IOSCAPU to capture a UCB failed. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ03132E UNCAPTURE FOR UCB FAILED; UCB
ADDRESS: nnnnnnnn R15: nnnn
TARGET VOLUME SERIAL: volser**

Explanation: An attempt to use IOSCAPU to uncapture a UCB failed. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ03133E EXCP FAILED FOR DEVICE: nnnn
TARGET VOLSER: volser SYNAD
TEXT: text**

Explanation: An error occurred reading the volume label for an offline target device. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ03134E VOLUME SERIAL FOR DEVICE: nnnn
IS existing volser; VOLUME SERIAL
SHOULD BE: paired source volser**

Explanation: The volume serial number for the indicated device is 'existing volser'. Based on the volume pairs specified, the expected volume serial number is 'paired source volser'. Processing terminates.

User response: Correct the volume pairs specified.

**CKZ03135E ICKDSF COMMAND FAILED FOR
TARGET VOLSER: volser**

Explanation: The invocation of ICKDSF to change a device label failed. The messages from ICKDSF are printed.

User response: If unable to determine the reason for the failure from the associated ICKDSF messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ03136W IEEVARYD VARY ONLINE FAILED
FOR TARGET VOLSER: volser
INVALID PARAMETERS**

Explanation: The parameters given to IEEVARYD are incorrect. Processing continues.

User response: Contact IBM Software Support. Have available the listing that contains this message. The indicated target volume serial will need to be manually varied online in order to proceed with DB2 Cloning Tool RENAME processing.

**CKZ03137W IEEVARYD VARY ONLINE FAILED
FOR TARGET VOLSER: volser
DEVICE: nnnn RETURN CODE:
nnnnnnnn REASON CODE: nnnnnnnn**

Explanation: The vary online for the indicated device failed. Processing continues.

User response: When the problem that caused the vary to fail is corrected, the indicated target volume serial will need to be manually varied online in order to proceed with DB2 Cloning Tool RENAME processing.

**CKZ03138W IEEVARYD VARY ONLINE FAILED
FOR TARGET VOLSER: volser
DEVICE: nnnn R15: nnnnnnnn**

Explanation: The vary online for the indicated device failed. Processing continues.

User response: When the problem that caused the vary to fail is corrected, the indicated target volume serial will need to be manually varied online in order to proceed with DB2 Cloning Tool RENAME processing.

**CKZ03140I DEVICE NUMBER: nnnn
SUCCESSFULLY CHANGED TO
VOLUME SERIAL: volser**

Explanation: The label of the indicated device has been changed by ICKDSF to the indicated volume serial number.

User response: No action is required.

CKZ03141I VOLUME SERIAL: volser DEVICE
NUMBER: nnnn IS NOW ONLINE

Explanation: The indicated target volume is now online to the current image.

User response: No action is required.

CKZ03142I DEVICE NUMBER: nnnn IS ONLINE
AS TARGET VOLUME SERIAL: volser

Explanation: The indicated device is currently online with the target volume serial.

User response: No action is required.

CKZ03143E DEVICE NUMBER: nnnn IS ONLINE
AS SOURCE VOLUME SERIAL: volser

Explanation: The indicated device is currently online with the source volume serial. The device is expected to be either offline or online with the target volume serial. Processing terminates.

User response: The device should be taken offline and the command rerun.

CKZ03144E DEVICE NUMBER: nnnn HAS
UNEXPECTED VOLUME SERIAL:
volser

Explanation: The indicated device is currently online with a volume serial that is neither the source or target volume serial. The device is expected to have a volume serial that is either the source or target volume serial. Processing terminates.

User response: Verify the device has not been overlaid with the contents of the wrong volume.

CKZ03145I DEVICE NUMBER: nnnn ALREADY
CHANGED TO TARGET VOLUME
SERIAL: volser

Explanation: The indicated device currently has the desired target volume serial.

User response: No action is required.

CKZ03146I DEVICE NUMBER: nnnn ICKDSF
FAILED; DEVICE MAY HAVE
ALREADY BEEN CLIPPED

Explanation: The clip of the indicated device by ICKDSF has failed. This failure is probably caused by the device having already been clipped.

User response: See the next DB2 Cloning Tool message in the listing to determine the appropriate action.

CKZ03148I VOLUME SERIAL: vvvvvv DEVICE
NUMBER: nnnn NOT CLIPPED TO
VOLUME SERIAL: vvvvvv DUE TO
SIMULATION

Explanation: The clip of the indicated device was not done because this run is a simulation.

User response: No action is required.

CKZ03501I hh:mm:ss CHECK USERCATALOGS
STARTED - PROGRAM REV=rrr |
hh:mm:ss CHECK USERCATALOGS
COMPLETED; RETURN CODE=nnn

Explanation: CHECK USERCATALOGS processing message.

User response: No action is required.

CKZ03503I DDNAME=ddname ALLOCATED FOR
DSN=datasetname

Explanation: 'ddname' has been dynamically allocated for the indicated data set.

User response: No action is required.

CKZ03505W DEALLOCATION FAILED FOR
DDNAME: ddname

Explanation: Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response: No action is required.

CKZ03507W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

CKZ03559E ALLOCATION FOR USERCATALOG
FAILED; DSN=bcs dsname

Explanation: The BCS dsname was not successfully allocated for further checking.

User response: Check that the user catalogs have been specified correctly.

CKZ03560I WAITING FOR SHARED CONTROL
OF BCS bcs name

Explanation: The BCS is currently in use by another job. The wait will continue until the BCS is no longer in use by another job. Processing continues.

User response: No action is required.

CKZ03561E UNABLE TO ALLOCATE BCS: bcs name; WAIT TIME LIMIT EXCEEDED

Explanation: The wait for shared control of the BCS has exceeded the wait time limit. Processing terminates.

User response: Change the scheduling of the jobs so the DB2 Cloning Tool job does not run when another job has the BCS allocated. Or increase the wait time limit so the DB2 Cloning Tool job can wait longer for the other job to terminate. The wait time limit is set by the CKZINI parameter CONCURRENT_EXECUTIONS_WAIT_TIME.

CKZ03570E SECURITY PRODUCT DENIED ACCESS TO DSN: datasetname

Explanation: The dsn indicated is not authorized for alter by your security product. If the RACF profile that is associated is returned, it will be displayed. Processing terminates.

User response: Change the dsn to one you can use, or, have your security administrator give you 'ALTER' authority to the data set.

CKZ03571E RACROUTE ERROR; SAF RC=nnnn RACF RC=nnnn RACF REASON CODE=nnnn

Explanation: An unexpected return code from SAF or RACF occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing containing this message.

CKZ04001I hh:mm:ss BCS BACKUP STARTED - PROGRAM REV=rrr | hh:mm:ss BCS BACKUP COMPLETED; RETURN CODE=nnn

Explanation: BCS BACKUP processing message.

User response: No action is required.

CKZ04003I DDNAME=ddname ALLOCATED FOR DSN=datasetname

Explanation: 'ddname' has been dynamically allocated for the indicated data set.

User response: No action is required.

CKZ04004E OPEN FAILED FOR DDNAME=ddname

Explanation: 'ddname' was allocated for DB2 Cloning Tool to use, but, the open for the file failed. Processing terminates.

User response: If unable to determine the reason the open failed, contact IBM Software Support. Have available the listing that contains this message.

CKZ04005E ALLOCATION FAILED FOR DSN: datasetname | ALLOCATION FAILED FOR DDNAME: ddname

Explanation: Dynamic allocation for a data set or ddname failed. The associated z/OS messages are displayed. Processing terminates.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ04005W DEALLOCATION FAILED FOR DDNAME: ddname

Explanation: Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ04007W ERROR CALLING CKZ01HEX; FUNCTION: function R15=nnnn

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

CKZ04008E UNABLE TO LOAD PROGRAM: program name

Explanation: The indicated program name was not found. Processing terminates.

User response: Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

CKZ04030E AMSOPEN FAILED; R15=nnnn

Explanation: An attempt was made to issue an IDCAMS command. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ04031I NON-ZERO RETURNED BY IDCAMS; RC=nnnn

Explanation: An IDCAMS command failed with return code nnnn. The IDCAMS messages are displayed. Processing will continue if the IDCAMS return code is less than or equal to 4. Processing will terminate if the IDCAMS return code is greater than 4 or if IDCAMS returned more than 50 errors.

User response: If the IDCAMS return code was

greater than 8, there is a problem with the source user catalog that will probably prevent DB2 Cloning Tool backing it up. Correct the problem before invoking DB2 Cloning Tool COPY.

CKZ04033I IDCAMS VERIFY FOR BCS WAS SUCCESSFUL

Explanation: An IDCAMS VERIFY was issued for the BCS to be backed up. The VERIFY completed normally.

User response: No action is required.

CKZ04034I IDCAMS EXAMINE INDEXTST FOR BCS WAS SUCCESSFUL

Explanation: An IDCAMS EXAMINE INDEXTST was issued for the BCS to be backed up. The EXAMINE completed normally.

User response: No action is required.

CKZ04035I NUMBER OF RECORDS READ FROM BCS: nnnnnnnn

Explanation: The indicated number of records were read from the BCS.

User response: No action is required.

CKZ04036I NUMBER OF RECORDS WRITTEN TO BACKUP: nnnnnnnn

Explanation: The indicated number of records were written to the BCS backup file.

User response: No action is required.

CKZ04040W BCS RECORD HAS BEEN SKIPPED; LENGTH MISMATCH; VSAM RECORD LENGTH: nnnnnnnn BCS RECORD LENGTH: nnnnnnnn

Explanation: There is a mismatch in records lengths. VSAM READ returned a length that was different than the length indicated in the BCS record. The BCS record is printed, but, because of this length error, the BCS record is not written to the backup file.

User response: No action is required. But, this does indicate a problem with the entry in the source usercatalog.

CKZ04045E ERROR ACCESSING BCS=bcname; LOC=lllll

Explanation: A VSAM error occurred accessing the indicated BCS. Processing terminates.

User response: See associated CKZERRnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ04046E ERROR CALLING CKZ00045; FUNCTION: function R15=nnnn

Explanation: An error occurred using CKZ00045 to read a source catalog. Processing terminates.

User response: See associated CKZnnnnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ04047W READING OF SOURCE USERCATALOG BYPASSED DUE TO SIMULATION AND SOURCE VOLUME BEING OFFLINE VOLSER=volser BCS=catalogname

Explanation: The source catalog cannot be read because this is a simulation run and the source volume is not online. For a simulation run the source catalog is read from the source volume. Processing continues.

User response: None, unless a RENAME simulation is desired. To have RENAME SIM process with this catalogs entries either bring the source volume online for COPY SIM or run COPY without SIM.

CKZ04060I WAITING FOR EXCLUSIVE CONTROL OF BCS bcname

Explanation: The BCS is currently in use by another job. The wait will continue until the BCS is no longer in use by another job or the wait time limit is exceeded. Processing continues.

User response: No action is required.

CKZ04061E UNABLE TO ALLOCATE BCS: bcname; WAIT TIME LIMIT EXCEEDED

Explanation: The wait for exclusive control of the BCS has exceeded the wait time limit. Processing terminates.

User response: Change the scheduling of the jobs so the DB2 Cloning Tool job does not run when another job has the BCS allocated. Or increase the wait time limit so the DB2 Cloning Tool job can wait longer for the other job to terminate. The wait time limit is set by the CKZINI parameter CONCURRENT_EXECUTIONS_WAIT_TIME.

CKZ04070E ERROR DURING UCBLOOK FOR VOLSER=volume - RETURN CODE=nnnn REASON CODE=nnnn

Explanation: An error occurred using the UCBLOOK macro. Processing terminates.

User response: If unable to determine the cause of the error, contact IBM Software Support. Have available the listing containing this message.

CKZ04501I CKZ00045 PROGRAM REV=x

Explanation: Program CKZ00045 displaying its versioning information.

User response: No action is required.

CKZ04502E Requested catalog xxx not found on volume vvv

Explanation: The specified catalog was not found on the volume. Processing terminates.

User response: Correct USERCATALOGS parameters to specify the source volser where the catalog resides.

CKZ04513E VOLSER xxx not found online

Explanation: VOLSER was not found on the system. Processing terminates.

User response: Correct the JOB's volser and resubmit.

CKZ04515E UNRECOGNIZED DEVICE TYPE: devicetype

Explanation: The UCBTYP for to volume is not defined as DASD. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ04517E Error processing VVCNs: message

Explanation: An error occurred during VVDS processing. Processing terminates.

User response: Contact IBM Software Support. Have available the listing containing this message.

CKZ04521E UCBLLOOK ERROR FOR VOLSER=volser RETURN CODE=nn REASON CODE=nn

Explanation: An error occurred during UCBLLOOK processing. Processing terminates.

User response: Contact IBM Software Support. Have available the listing containing this message.

CKZ04572I Accessing catalog xxx on volume yyy

Explanation: The identified catalog is being accessed.

User response: No action is required.

CKZ04573I Closing catalog xxx

Explanation: The identified catalog is no longer being accessed.

User response: No action is required.

CKZ04584E XXSETR FAILED

Explanation: Initialization of the catalog processor failed.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ04621E APF AUTHORIZATION FAILURE.

Explanation: The EXCP access routine requires APF authorization. Processing terminates.

User response: The load library being used (or one in the concatenation) is not APF authorized. Ensure all specified load libraries are authorized on that LPAR.

CKZ04624E LDS DATASETS NOT SUPPORTED

Explanation: An attempt was made to process a LDS using EXCPMODE. LDS support is not anticipated. Processing terminates.

User response: Only process LDS using record mode technology, not EXCP. This message should not occur, as the invoking code should be aware of the EXCPMODE inability.

CKZ04626W DSN=component not found on volser VTOC.

Explanation: The Format-1 DSCB for the VSAM component was not found on the VTOC. EXCP processing will attempt to use the data set extents carried in the VVR, instead. However, I/O errors may occur if the physical data on the tracks has been reused by another data set.

User response: No action is required.

CKZ04627E ERROR CALLING CKZ01VV1 TABLE=tablename FUNC=function R15=register15 R0=register0 LOC=location

Explanation: An error occurred processing an internal table. Processing terminates.

User response: Call IBM Software Support.

CKZ04628E UCB CAPTURE FAILED FOR VOLUME volser, ADR=ucbaddress, RC='returncode', RSN='reasoncode'

Explanation: An error occurred will attempting to acquire the UCB for the specified volume. Processing terminates.

User response: Call IBM Software Support.

CKZ04629E VVR IS MISSING
volumeinformationcell DSN=dsname

Explanation: The VSAM Volume Record for the specified DSN was not located in the VVDS. Processing terminates.

User response: Ensure the validity of VVDS by executing a DIAGNOSE. If the problem persists call IBM Software Support, for assistance.

CKZ04631E UCBLOOK FAILED FOR VOLUME
volser, RC='returncode',
RSN='reasoncode'

Explanation: An error occurred will attempting to pin the UCB for the specified volume. Processing terminates.

User response: Call IBM Software Support.

CKZ04632E DSN=dsname NOT FOUND -
module(returncode - reason code)

Explanation: An error occurred will attempting to retrieve volume data from the catalog. Processing terminates.

User response: Call IBM Software Support.

CKZ04633A DEBLOCK ERROR(code) volser
CCHHR=cyl_head_record
RDF_OFFSET=offset

Explanation: An error occurred while deblocking a control interval. Processing continues, but certain affected logical records may be unrecoverable. The CCHHR value indicates the physical block that is in error on the DASD device. The physical contents of that track can be displayed using the ADRDSSU PRINT command. ADRDSSU command example: PRINT TRACKS (X'cccc',X'hh',X'cccc',X'hh') INDY(volser)

User response: Call IBM Software Support. Provide the sysout of the failing job, including the Joblog output. Tech Support may also request the ADRDSSU output for the PRINT TRACKS command.

CKZ04634A Invalid CIDF Data X'????????' volser
CCHHR=cccc_hhhh_rr

Explanation: A control interval failed validation and could not be deblocked. Processing continues, but certain affected logical records may be unrecoverable. The CCHHR value indicates the physical block that is in error on the DASD device. The physical contents of that track can be displayed using the ADRDSSU PRINT command. ADRDSSU command example: PRINT TRACKS (X'cccc',X'hh',X'cccc',X'hh') INDY(volser)

User response: Call IBM Software Support. Provide the sysout of the failing job, including the Joblog output. Tech Support may also request the ADRDSSU

output for the PRINT TRACKS command.

CKZ04635A SPANNED RECORD ERROR - volser
CCHHR=cccc_hhhh_rr

Explanation: While processing a spanned record, an error occurred. Spanned records, for all but ESDS, must be re-assembled by using the data and the index entry - specifically the sequence of segments is controlled by the FLP in the index entry. The Volser and CCHHR value represents the physical address of the beginning of the Control Interval that is in error. Processing continues.

User response: Refer to other associated messages issued in conjunction with this error.

CKZ04636W TRUNCATED RECORD -
RBA=X'xxxxxxxx.xxxxxxx'

Explanation: While processing a spanned record, an error occurred. Processing continues.

User response: Refer to other associated messages issued in conjunction with this error.

CKZ04638E EXTENDED-FORMAT STRIPED
DATASETS NOT SUPPORTED.

Explanation: An attempt was made to process a striped object in EXCPMODE. This is not supported. Processing terminates.

User response: Call IBM Software Support.

CKZ04639E EXTENDED-FORMAT COMPRESSED
DATASETS NOT SUPPORTED.

Explanation: An attempt was made to process a compressed object in EXCPMODE. This is not supported. Processing terminates.

User response: Call IBM Software Support.

CKZ04640E DATASET NOT FOUND OR NOT
SPECIFIED

Explanation: The EXCPMODE interface was incorrectly used. Processing terminates.

User response: Call IBM Software Support.

CKZ04641E INVALID XXSET CALL

Explanation: The XXSET macro had one or more incorrectly specified arguments. Processing terminates.

User response: Call IBM Software Support.

CKZ04642E VOLSER NOT SPECIFIED

Explanation: The XXSET macro had no VOLSER argument coded. Processing terminates.

User response: Call IBM Software Support.

CKZ04643E INVALID VOLSEQ VALUE SPECIFIED

Explanation: The XXSET macro had an invalid SEQ argument coded. Processing terminates.

User response: Call IBM Software Support.

**CKZ04644E DUPLICATE XXSET FOR component
DSN=datasetname VOL=volser
SEQ=sequence#**

Explanation: Multiple XXSET macros with the same arguments were processed. Processing terminates.

User response: Call IBM Software Support.

**CKZ04645E First VOLSEQ Missing for Data
Component DSN=comp.name**

Explanation: The first volume of a multi-volume set for the Data Component was not identified. Processing cannot continue because necessary information such as the C/I Size cannot be determined. Processing terminates.

User response: Contact IBM Software Support. Provide the sysout listing for the JOB execution, including the Joblog. IBM Software Support may ask for additional listings from various utilities to identify the problem.

**CKZ04645W First VOLSEQ Missing for Index
Component DSN=comp.name**

Explanation: The first volume of a multi-volume set for the Index Component was not identified. Processing continues without the Index Component to assist in reassembling spanned logical records. However, if any spanned records are encountered, deblocking errors are likely to occur.

User response: No action is required.

**CKZ04646E First Data Component VVR not a "Z"
record, DSN=comp.name**

Explanation: The first volume's VVR for the Data Component was an unexpected type. Processing cannot continue because necessary information such as the C/I Size cannot be determined. Processing terminates.

User response: Contact IBM Software Support. Provide the sysout listing for the JOB execution, including the Joblog. IBM Software Support may ask for additional listings from various utilities to identify the problem.

**CKZ04646W First Index Component VVR not a "Z"
record, DSN=comp.name**

Explanation: The first volume's VVR for the Index Component was an unexpected type. Processing continues without the Index Component to assist in reassembling spanned logical records. However, if any spanned records are encountered, deblocking errors are likely to occur.

User response: No action is required.

**CKZ04647E EXCP Error - cmd descr volser cc_hh_r
dsn**

Explanation: An EXCP error occurred while reading the data set. Information is extracted from the standard IBM SYNAD Message. It is likely that the data set has been physically corrupted. Processing terminates.

User response: Contact IBM Software Support. Provide the sysout listing for the JOB execution, including the Joblog. IBM Software Support may ask for additional listings from various utilities to identify the problem.

**CKZ04647W EXCP Error - cmd descr volser cc_hh_r
dsn**

Explanation: An EXCP error occurred while reading the data set. Information is extracted from the standard IBM SYNAD Message. It is likely that the data set has been physically corrupted. The nature of this error is not considered immediately critical. Processing continues.

User response: No action is required. Informational warning message.

CKZ04700I PGM CKZ00047 PROGRAM REV=x

Explanation: Program CKZ00047 displaying its versioning information.

User response: No action is required.

**CKZ04701E CKZ00047 Parameter Error. parameter
description**

Explanation: CKZ00047 has detected invalid parameters from the calling routine. This is an internal error.

User response: Contact IBM Software Support. Have the execution output listing available.

**CKZ04702I PGM CKZ00047 invoked to perform
???????? function on VOL=volser
UNIT=addr - PROGRAM REV=rrr**

Explanation: Program CKZ00047 is acknowledging a request to DUMP or RESTORE the VTOC and VVDS of the indicated volume.

CKZ04704E • CKZ04749E

User response: No action is required.

CKZ04704E DSPSERV CREATE Error RC=xx,RSN=yy, requesting nnnnn 4K Dataspace Blocks.

Explanation: Dataspace Creation failed with the above Return and Reason codes.

User response: Contact IBM Software Support. Have the execution output listing available.

CKZ04705E ALESERV ADD Error R15=xx Creating Dataspace Alet

Explanation: An error occurred while attempting to add an entry into the DU-AL for a private dataspace that has been created.

User response: Contact IBM Software Support. Have the execution output listing available.

CKZ04706I PGM CKZ00047 ???????? Processing Completed RC=xx timestamp

Explanation: Program CKZ00047 processing is terminating with the above return-code.

User response: If RC=00, None. If the Return-Code is any non-zero value, then contact IBM Software Support, and have the execution output listing available. There will be previous messages indicating the error causing the bad return code.

CKZ04710E CKZ00047 ABENDED S-xxx | CKZ00047 ABENDED U-xxxx

Explanation: Program CKZ00047 has suffered an abend and is taking appropriate recovery and cleanup actions. The requested function appearing in the CKZ04700I message has failed.

User response: Contact IBM Software Support. Have the execution listings and the SYSUDUMP output available. Also, make note of the MVS operating system release, and the type of hardware that was being accessed.

CKZ04720E I/O Error Reading Volume Label on Device /xxxx

Explanation: Program CKZ00047 was unable to read the volume label at the indicated device address.

User response: Determine if the device at the indicated address can be varied OFFLINE and ONLINE. The volume may be uninitialized. If the volume can be successfully mounted, then contact IBM Software Support. Have the execution output listing available. Also, make note of the MVS operating system release, and the type of hardware that was being accessed.

CKZ04721E Validation on Device /xxxx failed. Detected VOL=yyyyyy

Explanation: Program CKZ00047 read the volume label at the indicated device address and found a volume serial number that was different than what was expected.

User response: Contact IBM Software Support. Have the execution output listing available.

CKZ04733E DataSpace size is Insufficient.

Explanation: While preparing to DUMP the VTOC and/or VVDS of the volume indicated by the CKZ04700I message, program CKZ00047 was unable to allocate a private dataspace of a sufficient size.

User response: Contact IBM Software Support. Have the execution output listing available. Also, make note of the MVS operating system release.

CKZ04747E Error Allocating ????? bytes for record ????? of ??? | Dataspace size ???-K is insufficient. | ???-K used up to this point.

Explanation: While preparing to RESTORE the VTOC and/or VVDS of the indicated volume, CKZ00047 exceeded a predetermined dataspace size.

User response: Contact IBM Software Support. Have the execution output listing available. If possible, save the backup data set indicated by this message in case it is requested by IBM Software Support.

CKZ04748E Invalid Dump Record. ID ????????

Explanation: While preparing to RESTORE the VTOC and/or VVDS of the indicated volume, CKZ00047 has determined that the logical contents of the sequential backup data set are invalid.

User response: Contact IBM Software Support. Have the execution output listing available. If possible, save the backup data set indicated by this message in case it is requested by IBM Software Support.

CKZ04749E Buffer Capacity Exceeded. TYPE=????

Explanation: RESTORE processing has failed due to incorrect buffer size calculations. This is an internal error.

User response: Contact IBM Software Support, and have the execution output listing available. Also, make note of the MVS operating system release, and the type of hardware that was being accessed.

CKZ04750E ** EXCP I/O ERROR processing the
????????? ** | Track(CCHH) Address:
cchh | Synad: SYNAD error text

Explanation: An uncorrectable error has occurred to the device being DUMPed or RESTORed while CKZ00047 was performing I/O using the EXCP access method.

User response: Contact IBM Software Support, and have the execution output listing available. Also, make note of the MVS operating system release, and the type of hardware that was being accessed.

CKZ04754E Device ???? - No Paths Available.
Reason: ?????? | Reason: IOS Message
text

Explanation: While attempting to access the device, CKZ00047 could not identify an available channel path to the device slated for DUMP or RESTORE processing.

User response: Verify that channel paths are available to device by issuing MVS display commands such as D M=DEV(xxxx) and D M=CHP(yy). If device pathing appears valid, then contact IBM Software Support, and have the execution output listing available. Also, make note of the MVS operating system release, and the type of hardware that was being accessed.

CKZ04755E Storage Subsystem for Device ???? Does
not Support ECKD CCWs.

Explanation: CKZ00047 has detected an old technology DASD Control Unit that does not support hardware features that are minimally required by this. Such control units are typically used for supporting devices that pre-date 3380's. All control units for 3390's support ECKD transfer protocol.

User response: No action is required. DB2 Cloning Tool cannot be used for this device.

CKZ04761I *Informational messages*

Explanation: Informational statistics regarding DUMP processing.

User response: No action is required.

CKZ04761W Note: Requested VVDS Dataset not in
use.

Explanation: Informational warning regarding DUMP processing. The caller of CKZ00047 specified a non-standard data set name for the VVDS, which was not found on the volume. The correct data set name for the VVDS was found, and will be assumed as valid for DUMP processing.

User response: No action is required. DUMP processing continues.

CKZ04761E ** ERROR ** Requested VVDS Dataset
Not Found. | ** ERROR ** Required
VVDS Dataset Not Found.

Explanation: A VVDS was not found on the volume that was being processed for dump. In addition, the volume was SMS managed, and/or contained VSAM data sets.

User response: Verify that the volume is usable. If not, then a volume restore is in order. In either case, contact IBM Software Support, and have the execution output listings available.

CKZ04807W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

CKZ04808E UNABLE TO LOAD PROGRAM:
program name

Explanation: The indicated program name was not found. Processing terminates.

User response: Check that the job's STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

CKZ04817E ERROR PROCESSING VVCNS:
message

Explanation: An error occurred during VVDS processing. Processing terminates.

User response: Contact IBM Software Support. Have available the listing containing this message.

CKZ04818E VOLSER: vvvvvv IS NOT ONLINE

Explanation: The indicated volume is not online. Processing terminates.

User response: Bring the volume online.

CKZ04860E UCBLOOK ERROR; RETURN
CODE=nn REASON CODE=nn
LOC=IIIII

Explanation: An error occurred during UCBLOOK processing. Processing terminates.

User response: Contact IBM Software Support. Have available the listing containing this message.

CKZ04861E IOSCAPU function ERROR; RETURN CODE=nn REASON CODE=nn LOC=IIIII

Explanation: An attempt to use IOSCAPU to capture or uncapture a UCB failed. Processing terminates.

User response: Contact IBM Software Support. Have available the listing containing this message.

CKZ05001I hh:mm:ss COPYSYNC STARTED - PROGRAM REV=rrr | hh:mm:ss COPYSYNC COMPLETED; RETURN CODE=nnn

Explanation: COPYSYNC processing message.

User response: No action is required.

CKZ05003I DDNAME=ddname ALLOCATED FOR DSN=datasetname

Explanation: 'ddname' has been dynamically allocated for the indicated data set.

User response: No action is required.

CKZ05004E DDNAME MISSING: ddname

Explanation: 'ddname' was specified for DB2 Cloning Tool to use, but, the ddname was not found in the JCL. Processing terminates.

User response: Compare the specification of the ddname with the JCL used for the command.

CKZ05005E ALLOCATION FAILED FOR DSN: datasetname | ALLOCATION FAILED FOR DDNAME: ddname

Explanation: Dynamic allocation for a data set or ddname failed. The associated z/OS messages are displayed. Processing terminates.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ05007W ERROR CALLING CKZ01HEX; FUNCTION: function R15=nnnn

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

CKZ05008E UNABLE TO LOAD PROGRAM: program name

Explanation: The indicated program name was not found. Processing terminates.

User response: Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

CKZ05009E ERROR ACCESSING JOURNAL FILE; LOC=IIIII

Explanation: A VSAM error occurred accessing the journal file. Processing terminates.

User response: See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ05011E JOURNAL CONTROL RECORD NOT FOUND | JOURNAL VOLUME PAIR RECORD(S) NOT FOUND

Explanation: An expected record was not found in the DB2 Cloning Tool journal file. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ05013E RECORD COUNT IS ZERO; LOC=IIIII | COUNT MISMATCH; RECORDS READ=rrrr CONTROL RECORD COUNT=cccc

Explanation: There was a problem with the journal records needed to initiate the command. For the first format, the journal control record indicates no entries were added. For the second format, the number of records read from the journal, rrrr, is not the same as the number indicated in the journal control record, cccc. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ05015W THE COPY COMMAND WAS NOT INITIATED BY A METHOD REQUIRING THE COPYSYNC COMMAND.

Explanation: Only a copy initiated by PPRC requires the COPYSYNC command. Processing terminates.

User response: If PPRC was used to initiated the volume copies, contact IBM Software Support. Have available the listing that contains this message.

CKZ05016E COPY STATUS IS n

Explanation: Only a copy initiated by PPRC requires the COPYSYNC command. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ05031I COPY STEP WAS A SIMULATION

Explanation: The journal indicates that the COPY command was a simulation. No volume synchronization checking will be done.

User response: No action is required.

**CKZ05051E REQUIRED KEYWORD MISSING:
keyword**

Explanation: A keyword required for processing has been omitted. Processing terminates.

User response: Specify the required keyword.

**CKZ05053E KEYWORD: keyword MAXIMUM
LENGTH: nnn EXCEEDED**

Explanation: The operand entered for a keyword exceeded the maximum length allowed for the operand. nnn is the maximum allowed length for the keyword. Processing terminates.

User response: Correct the length of the keyword's operand.

**CKZ05054E KEYWORD HAS MORE THAN 1
OPERAND; ONLY 1 ALLOWED:
keyword**

Explanation: Multiple operands were detected for a keyword; only one operand is permitted. Processing terminates.

User response: Correct the keyword to use one operand.

**CKZ05056E NOTHING SPECIFIED FOR
KEYWORD: keyword**

Explanation: A keyword was entered without an appropriate operand. Processing terminates.

User response: Specify an appropriate operand for the keyword.

**CKZ05058E INVALID VALUE IN KEYWORD:
keyword VALUE: value error text**

Explanation: The value in the keyword is invalid. 'error text' indicates the problem detected with the value. Processing terminates.

User response: Correct the value specified in the keyword.

**CKZ05101I hh:mm:ss PPRC COPYSYNC STARTED
- PROGRAM REV=rrr | hh:mm:ss
PPRC COPYSYNC COMPLETED;
RETURN CODE=nnn**

Explanation: PPRC COPYSYNC processing message.

User response: No action is required.

**CKZ05107W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn**

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

**CKZ05108E UNABLE TO LOAD PROGRAM:
program name**

Explanation: The indicated program name was not found. Processing terminates.

User response: Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

**CKZ05130I VOLUME PAIRS STATUS source
volser/target volser status**

Explanation: The status of each source/target volume serial pair is displayed.

User response: No action is required.

CKZ05141W TIME LIMIT EXCEEDED

Explanation: The WAIT time for the command has been reached. Some volume pairs may still be in a copy relationship.

User response: Rerun the COPYSYNC command with a higher WAIT time specified.

CKZ05143I ANTRQST LEVEL=nn

Explanation: The level returned by ANTRQST REQUEST=LEVEL

User response: No action is required.

CKZ05144E ANTRQST LEVEL NOT SUPPORTED

Explanation: For PPRC support, the level must be one (1). The level returned by ANTRQST is not supported. Processing terminates.

User response: Check with your system programmer for upgrading the system.

CKZ05145E system task NOT ACTIVE

Explanation: An ANTRQST request failed because a system task is not active. Processing terminates.

User response: Start the indicated system task.

CKZ05146E ANTRQST DID NOT RETURN ANY INFORMATION; RESTART system task SYSTEM TASK

Explanation: The ANTRQST REQUEST=PQUERY did not receive information for a device. Processing terminates.

User response: Restart the indicated system task.

CKZ05147E ANTRQST ERROR; request type RETURN CODE=nnnn X'hhhh' REASON CODE=nnnn X'hhhh' LOC=IIIII

Explanation: An ANTRQST request failed. 'request type' indicates if the macro was incorrect, or, if the Data Mover failed the request. The return code and reason code are displayed in decimal and hexadecimal format. 'IIIII' is an internal indicator of where the problem occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ05150E UNRECOGNIZED STATUS FROM PQUERY

Explanation: The device status returned by PQUERY was not recognized. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ05151I VOLSER PAIR source volser/target volser COPY STILL IN PROGRESS, nnn% COMPLETED

Explanation: PQUERY indicates the volumes are still in a copy relationship. nnn% indicates the percent of the copy that is completed.

User response: No action is required.

CKZ05152I VOLSER PAIR source volser/target volser PDELPAIR ISSUED

Explanation: PDELPAIR has been issued for the volumes.

User response: No action is required.

CKZ05501I hh:mm:ss VOLUME CHECK STARTED - PROGRAM REV=rrr | hh:mm:ss VOLUME CHECK COMPLETED; RETURN CODE=nnn

Explanation: COPYCHECK command processing message.

User response: No action is required.

CKZ05503I DDNAME=ddname ALLOCATED FOR DSN=datasetname

Explanation: 'ddname' has been dynamically allocated for the indicated data set.

User response: No action is required.

CKZ05504E DDNAME MISSING: ddname

Explanation: 'ddname' was specified for DB2 Cloning Tool to use. Processing terminates.

User response: Either correct the ddname specified, or, add the appropriate ddname to the job's JCL.

CKZ05505E ALLOCATION FAILED FOR DSN: datasetname

Explanation: Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ05505W DEALLOCATION FAILED FOR DDNAME: ddname

Explanation: Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ05507W ERROR CALLING CKZ01HEX; FUNCTION: function R15=nnnn

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

**CKZ05508E UNABLE TO LOAD PROGRAM:
program name**

Explanation: The indicated program name was not found. Processing terminates.

User response: Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

**CKZ05509E ERROR ACCESSING JOURNAL FILE;
LOC=IIIII**

Explanation: A VSAM error occurred accessing the journal file. Processing terminates.

User response: See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ05511E JOURNAL CONTROL RECORD NOT
FOUND | JOURNAL VOLUME PAIR
RECORD(S) NOT FOUND**

Explanation: An expected record was not found in the DB2 Cloning Tool journal file. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ05512E JOURNAL CONTROL RECORD IS
WRONG VERSION | JOURNAL VOLP
RECORD IS WRONG VERSION**

Explanation: The journal record does not match the expected format. The record is printed. Processing terminates.

User response: Verify that different releases of DB2 Cloning Tool have not been run using the same journal data set. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains this message.

**CKZ05513E RECORD COUNT IS ZERO; LOC=IIIII
| COUNT MISMATCH; RECORDS
READ=rrrr CONTROL RECORD
COUNT=cccc LOC=IIIII**

Explanation: There was a problem with the journal records needed to initiate the volume check. For the first format, the journal control record indicate no entries were added. For the second format, the number of records read from the journal, rrrr, is not the same as the number indicated in the journal control record, cccc. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ05515W THE COPY PROCESS DID NOT
COMPLETE SUCCESSFULLY**

Explanation: The journal indicates that the COPY command did not complete successfully. Processing continues.

User response: No action is required.

CKZ05531I COPY STEP WAS A SIMULATION

Explanation: The journal indicates that the COPY command was a simulation. No volume checking will be done.

User response: No action is required.

**CKZ05540W COPIES DONE OUTSIDE OF DB2
CLONING TOOL; NO VOLUME
CHECKING WILL BE DONE**

Explanation: The volume pairing was not initiated by the DB2 Cloning Tool COPY command.

User response: No action is required.

**CKZ05550E ERROR IN PARAMETERS FOR
keyword**

Explanation: The parameters for the indicated keyword were incorrect. Processing terminates.

User response: Check the keyword parameters. Mutually exclusive keywords may have been used.

**CKZ05551E REQUIRED KEYWORD MISSING:
keyword**

Explanation: A keyword required for processing has been omitted. Processing terminates.

User response: Specify the required keyword.

**CKZ05553E KEYWORD: keyword MAXIMUM
LENGTH: nnn EXCEEDED**

Explanation: The operand entered for a keyword exceeded the maximum length allowed for the operand. nnn is the maximum allowed length for the keyword. Processing terminates.

User response: Correct the length of the keyword's operand.

**CKZ05554E KEYWORD HAS MORE THAN 1
OPERAND; ONLY 1 ALLOWED:
keyword**

Explanation: Multiple operands were detected for a keyword; only one operand is permitted. Processing terminates.

User response: Correct the keyword to use one operand.

CKZ05556E NOTHING SPECIFIED FOR
KEYWORD: keyword

Explanation: A keyword was entered without an appropriate operand. Processing terminates.

User response: Specify an appropriate operand for the keyword.

CKZ05558E INVALID VALUE IN KEYWORD:
keyword VALUE: value error text

Explanation: The value in the keyword is invalid. 'error text' indicates the problem detected with the value. Processing terminates.

User response: Correct the value specified in the keyword.

CKZ05586I VALIDATING KEYWORD: keyword

Explanation: Parsing is checking the indicated keyword indicated in the command.

User response: No action is required.

CKZ05601I hh:mm:ss VOLUME STATUS STARTED
- PROGRAM REV=rrr | hh:mm:ss
VOLUME STATUS COMPLETED;
RETURN CODE=nnn

Explanation: Volume status processing message.

User response: No action is required.

CKZ05607W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

CKZ05608E UNABLE TO LOAD PROGRAM:
program name

Explanation: The indicated program name was not found. Processing terminates.

User response: Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

CKZ05630I VOLUME PAIRS STATUS source
volser/target volser status

Explanation: The status of each source/target volume serial pair is displayed.

User response: No action is required.

CKZ05638I UNEXPECTED RESULTS FROM
FCQUERY FQMAP; VOLSER= volser

Explanation: An ANTRQST FCQUERY FQMAP call returned unexpected results.

User response: Please report this message to IBM Software Support.

CKZ05641W TIME LIMIT EXCEEDED

Explanation: The WAIT time for the command has been reached. Some volume pairs may still be in a copy relationship.

User response: No action is required. If desired, rerun the COPYCHECK command with a higher WAIT time specified.

CKZ05643I ANTRQST LEVEL=nn; ESSRVCS
LEVEL=nnn

Explanation: The level returned by ANTRQST REQUEST=LEVEL

User response: No action is required.

CKZ05648E type VOLUME volser ATTRIBUTES
HAVE CHANGED;

Explanation: The indicated volume has physically moved since COPY. 'type' indicates if this is a SOURCE or TARGET volume. COPYCHECK is unable to process the moved volume. Processing terminates.

User response: Do not use COPYCHECK after SOURCE or TARGET volumes have been moved from where they were at the time of COPY.

CKZ05649E type VOLUME volser IS NOT ONLINE

Explanation: The indicated volume is not online. 'type' indicates if this is a SOURCE or TARGET volume. COPYCHECK is unable to process the volume. Processing terminates.

User response: Do not use COPYCHECK if SOURCE or TARGET volumes are offline.

CKZ05644E ANTRQST LEVEL NOT SUPPORTED;
LEVEL=nn

Explanation: For FlashCopy support, the level must be greater than four (4). For SnapShot support, the level must be greater than one (1). For PPRC support, the level must be one (1). The level returned by ANTRQST is not supported. Processing terminates.

User response: Check with your system programmer for upgrading the system.

CKZ05645E system task NOT ACTIVE

Explanation: An ANTRQST request failed because a 'system task' is not active. Processing terminates.

User response: Start the indicated system task.

CKZ05646E ANTRQST DID NOT RETURN ANY INFORMATION; RESTART system task SYSTEM TASK

Explanation: The ANTRQST REQUEST=FCQUERY did not receive information for a device, or, REQUEST=PQUERY did not receive information for a device. Processing terminates.

User response: Restart the indicated system task.

CKZ05647E ANTRQST ERROR; request type RETURN CODE=nnnn X'hhhh' REASON CODE=nnnn X'hhhh' LOC=lllll

Explanation: An ANTRQST request failed. 'request type' indicates if the macro was incorrect, or, if the Data Mover failed the request. The return code and reason code are displayed in decimal and hexadecimal format. 'lllll' is an internal indicator of where the problem occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ05650E UNRECOGNIZED STATUS FROM PQUERY

Explanation: The device status returned by the indicated query was not recognized. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ05650E UNRECOGNIZED STATUS FROM PQUERY

Explanation: The device status returned by the indicated query was not recognized. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ05651I VOLSER PAIR source volser/target volser COPY STILL IN PROGRESS, nnn% COMPLETED | VOLSER PAIR source volser/target volser COPY STILL IN PROGRESS, FCNOCOPY

Explanation: FCQUERY indicates the volumes are still in a copy relationship. If the background copy was initiated, nnn% will indicate the percent of the copy that is completed. If FCNOCOPY was used, no

background copy was initiated and the second form of the message will be issued.

User response: No action is required.

CKZ05652I VOLSER PAIR source volser/target volser FCWITHDRAW ISSUED | VOLSER PAIR source volser/target volser PDELPAIR ISSUED

Explanation: FCWITHDRAW or PDELPAIR has been issued for the volumes.

User response: No action is required.

CKZ05653I VOLSER pair: source volser//target volser FCWITHDRAW indicated the relationship no longer existed

Explanation: The FCWITHDRAW for the volumes indicated that there was no FlashCopy relationship between the two volumes. There is a small window between the check for FlashCopy relationships on the volumes and when the FlashCopy WITHDRAW is done. If the relationship ends between the check and FlashCopy WITHDRAW, this message is displayed to document the event.

User response: No action is required.

CKZ05660E UCBLOOK ERROR; RETURN CODE=nn REASON CODE=nn

Explanation: An error occurred using the UCBLOOK macro. Processing terminates.

User response: Contact IBM Software Support. Have available the listing containing this message.

CKZ05701I hh:mm:ss EMC VOLUME STATUS STARTED - PROGRAM REV=rrr | hh:mm:ss EMC VOLUME STATUS COMPLETED; RETURN CODE=nnn

Explanation: EMC volume status processing message.

User response: No action is required.

CKZ05707W ERROR CALLING CKZ01HEX; FUNCTION: function R15=nnnn

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

CKZ05708E UNABLE TO LOAD PROGRAM: program name

Explanation: The indicated program name was not found. Processing terminates.

User response: Check that the job's //STEPLIB library

is correct. If unable to resolve the problem, contact IBM Software Support.

**CKZ05730I VOLUME PAIRS STATUS source
volser/target volser status**

Explanation: The status of each source/target volume serial pair is displayed.

User response: No action is required.

**CKZ05738E VOLP ENTRY NOT FOUND FOR
VOLSER=volser**

Explanation: An internal processing error has occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ05739I EMCSNAPI COMPLETED; RETURN
CODE=nnn**

Explanation: A call to the EMCSNAPI module has completed with a return code of nnn.

User response: No action is required.

**CKZ05740E INTERNAL ERROR; DATA MOVER
SETTING X'dd' INVALID**

Explanation: An internal processing error has occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ05741W TIME LIMIT EXCEEDED

Explanation: The WAIT time for the command has been reached. Some volume pairs may still be in a snap relationship.

User response: No action is required. If desired, rerun the COPYCHECK command with a higher WAIT time specified.

**CKZ05742W STOP SNAP PENDING; SNAPS ARE
STILL ACTIVE**

Explanation: A STOPSNAP was requested and 10 minutes after the STOP SNAP commands have been issued there are still some volume pairs in a snap relationship.

User response: Determine why the snap relationships have not stopped.

CKZ05743I module-name VERSION version

Explanation: This message reports the version information retrieved from the indicated module.

User response: No action is required.

CKZ05745E EMCSCF NOT ACTIVE; LOC=IIII

Explanation: An EMC request failed because the system task, EMCSCF, is not active. Processing terminates.

User response: Start the system task, EMCSCF.

**CKZ05746E module-name ERROR; request type
R15=nnnn X'hhhh' LOC=IIII
VOLSER=volser**

Explanation: The call to the indicated module for the indicated request type failed. The return code is displayed in decimal and hexadecimal format. 'IIII' is an internal indicator of where the problem occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ05747E module-name ERROR; request type
RETURN CODE=nnnn X'hhhh'
REASON CODE=nnnn X'hhhh'
LOC=IIII VOLSER=volser**

Explanation: The call to the indicated module for the indicated request type failed. The return code and reason code are displayed in decimal and hexadecimal format. 'IIII' is an internal indicator of where the problem occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ05748E type VOLUME volser ATTRIBUTES
HAVE CHANGED; attributes**

Explanation: The indicated volume has physically moved since COPY. 'type' indicates if this is a SOURCE or TARGET volume. COPYCHECK is unable to process the moved volume. Processing terminates.

User response: Do not use COPYCHECK after SOURCE or TARGET volumes have been moved from where they were at the time of COPY.

**CKZ05749E type VOLUME volser IS NOT ONLINE
attributes**

Explanation: The indicated volume is not online. 'type' indicates if this is a SOURCE or TARGET volume. COPYCHECK is unable to process the volume. Processing terminates.

User response: Do not use COPYCHECK if SOURCE

or TARGET volumes are offline.

CKZ05751I VOLSER PAIR source volser/target
volser SNAP STILL IN PROGRESS,
nnn% COMPLETED - type

Explanation: The volumes are still in a snap relationship. 'nnn%' indicates the percent of the copy that is completed. 'type' indicates if this is a DIFFERENTIAL snap or a snap with no background copy.

User response: No action is required.

CKZ05752I VOLSER PAIR source volser/target
volser STOP SNAP ISSUED

Explanation: STOP SNAP has been issued for the volumes.

User response: No action is required.

CKZ05756I CURRENT SNAP RELATIONSHIPS
FOR type VOLSER: volser volser
relationship information

Explanation: The current snap relationships for the volser are listed. 'type' indicates if the volser is a SOURCE or TARGET volume.

User response: No action is required.

CKZ05760E UCBLOOK ERROR; RETURN
CODE=nn REASON CODE=nn

Explanation: An error occurred using the UCBLOOK macro. Processing terminates.

User response: Contact IBM Software Support. Have available the listing containing this message.

CKZ06001I hh:mm:ss BACKINFO REFORMAT
STARTED - PROGRAM REV=rrr |
hh:mm:ss BACKINFO REFORMAT
COMPLETED; RETURN CODE=nnn

Explanation: BACKINFO-REFORMAT processing message.

User response: No action is required.

CKZ06004E DDNAME MISSING: ddname | OPEN
FAILED FOR DDNAME: ddname

Explanation: ddname was specified for DB2 Cloning Tool to use. Processing terminates.

User response: Either correct the ddname specified, or, add the appropriate ddname to the job's JCL.

CKZ06006E ERROR CALLING CKZ01VV1 tttttt
FUNCTION: function R15=nnnn
R0=nnnnnnnn LOC=lllll

Explanation: A problem occurred using a dataspace. tttttt is the name of the internal table. llll is the location where the error occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message and the parmlib member that controls execution of DB2 Cloning Tool.

CKZ06007W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

CKZ06008E UNABLE TO LOAD PROGRAM:
program name

Explanation: The indicated program name was not found. Processing terminates.

User response: Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

CKZ06019E THE keyword DOES NOT HAVE A
LRECL OF 80, DDNAME: ddname

Explanation: The data set allocated to the ddname does not have a LRECL of 80. The LRECL of this data set must be 80. Processing terminates.

User response: Change the data set allocated to the ddname to have a LRECL of 80.

CKZ06020E UNKNOWN RECORD TYPE FOUND
IN BACKINFO record

Explanation: An unknown record type was found in the backinfo data set. Processing terminates.

User response: Check that a correct backinfo data set is being used. If unable to resolve the problem, contact IBM Software Support.

CKZ06021E DUPLICATE type VOLSER FOUND IN
BACKINFO; VOLSER: vvvvvv

Explanation: The indicated volser was found multiple times in the backinfo data set. Processing terminates.

User response: Check that a correct backinfo data set is being used. If unable to resolve the problem, contact IBM Software Support.

CKZ06022E DUPLICATE CATALOG DSN FOUND IN BACKINFO; DSN: dataset

Explanation: The indicated catalog DSN was found multiple times in the backinfo data set. Processing terminates.

User response: Check that a correct backinfo data set is being used. If unable to resolve the problem, contact IBM Software Support.

CKZ06023E NO VOLMAP RECORDS FOUND IN BACKINFO

Explanation: No VOLMAP type records were found in the backinfo data set. Processing terminates.

User response: Check that a correct backinfo data set is being used. If unable to resolve the problem, contact IBM Software Support.

CKZ06024E INVALID type DEVN FOUND IN BACKINFO; DEVN: nnnn reason

Explanation: An invalid device number was found in the backinfo data set. Processing terminates.

User response: Check that a correct backinfo data set is being used. If unable to resolve the problem, contact IBM Software Support.

CKZ06025E BACKUP VOLSER: vvvvvv IS NOT ONLINE

Explanation: The indicated backup volume volser is not online. Processing terminates.

User response: Bring the indicated volume online or use CLIP-IF-OFFLINE(Y) if the volume is offline and needs to be clipped.

CKZ06026E BACKUP VOLSER: vvvvvv IS NOT ONLINE BUT ITS SPECIFIED DEVN: dddd IS ONLINE WITH VOLSER: vvvvvv

Explanation: The indicated backup volume volser is not online, but its specified device number is online with a different volser. Processing terminates.

User response: Check that a correct backinfo data set is being used. If unable to resolve the problem, contact IBM Software Support.

CKZ06027E SPECIFIED BACKUP VOLSER: vvvvvv IS ONLINE ON DEVN: dddd WHICH IS NOT ITS SPECIFIED DEVN: dddd

Explanation: The indicated backup volume volser is online, but it is on a different device number than specified in the backinfo data set. Processing terminates.

User response: Check that a correct backinfo data set is being used. If unable to resolve the problem, contact IBM Software Support.

CKZ06028E UNABLE TO GENERATE BACKUP VOLSER FOR SOURCE VOLSER: vvvvvv; NO MATCHING VOLSER RENAME MASK FOUND

Explanation: A backup volume volser needs to be generated, but there is no entry in the VOLSER-RENAME-MASKS keyword that matches the volser of its corresponding source volume. Processing terminates.

User response: Add a entry to the VOLSER-RENAME-MASKS keyword that matches the indicated source volser.

CKZ06029E SPECIFIED BACKUP DEVN: dddd IS NOT DEFINED TO Z/OS

Explanation: The device number specified for a backup volume in the backinfo data set is not defined to z/OS. Processing terminates.

User response: Check that the device number specified is correct and the job is running on a z/OS system where the device is defined.

CKZ06030E BACKUP VOLSER NOT SPECIFIED FOR SOURCE VOLSER: vvvvvv

Explanation: A unique volser was not specified for a backup volume in the backinfo data set and the CLIP-IF-OFFLINE(Y) keyword was not used. Processing terminates.

User response: Determine if the CLIP-IF-OFFLINE(Y) keyword should be used or correct the backinfo data set to have a unique volser for the backup volume.

CKZ06031E DUPLICATE BACKUP VOLSER: vvvvvv FOUND FOR SOURCE VOLSER: vvvvvv AND SOURCE VOLSER: vvvvvv

Explanation: The backinfo data set has two VOLMAP records for different source volumes that have the same backup volser specified. Processing terminates.

User response: Check that a correct backinfo data set is being used. If unable to resolve the problem, contact IBM Software Support.

CKZ06032E SOURCE VOLSER: vvvvvv IS ALSO USED AS A BACKUP VOLSER FOR SOURCE VOLSER: vvvvvv

Explanation: The backinfo data set has a source volser that is also used as a backup volser for a different source volume. Processing terminates.

User response: Check that a correct backinfo data set is being used. If unable to resolve the problem, contact IBM Software Support.

**CKZ06033E VOLSER: vvvvvv FOR UCAT:
usercatalog DOES NOT MATCH ANY
SOURCE VOLSER**

Explanation: The backinfo data set has a UCAT record for the indicated user catalog, but the volser specified for it is not a source volser in a VOLMAP record. Processing terminates.

User response: Check that a correct backinfo data set is being used. If unable to resolve the problem, contact IBM Software Support.

**CKZ06034E USERCATALOGS ENTRY: usercatalog
DOES NOT MATCH ANY UCAT
ENTRY FOUND IN BACKINFO**

Explanation: The indicated user catalog was specified in the USERCATALOGS keyword, but the backinfo data set does not have a UCAT entry for that user catalog. Processing terminates.

User response: Check that a correct backinfo data set is being used and the user catalog has been specified correctly in the USERCATALOGS keyword.

**CKZ06035I NO BACKUP VOLUMES NEED TO BE
CLIPPED**

Explanation: All the backup volumes are already online with their expected volsers.

User response: No action is required.

**CKZ06036E NO BACKUP VOLSER SPECIFIED IN
BACKINFO FOR SOURCE VOLSER:
vvvvvv BUT ITS SPECIFIED DEVN:
dddd IS ONLINE WITH VOLSER:
vvvvvv**

Explanation: The VOLMAP record in the backinfo data set for source volser has no backup volume volser. This condition means that the backup volume is expected to be offline but the backup volume device was found to be online with the indicated volser. Processing terminates.

User response: Check that a correct backinfo data set is being used. If this is a rerun of a prior failed CLIP-IF-OFFLINE(Y) run, specify the RESUME keyword.

**CKZ06037I BACKUP VOLUME ON DEVN: dddd
APPEARS TO HAVE ALREADY BEEN
CLIPPED TO VOLSER: vvvvvv**

Explanation: The CLIP-IF-OFFLINE(Y) and RESUME keywords have been specified and the indicated

backup volume appears to have already been clipped to the indicated volser.

User response: No action is required.

**CKZ06038I BACKUP VOLUME ON DEVN: dddd
WILL BE CLIPPED TO {GENERATED |
SPECIFIED} VOLSER: vvvvvv**

Explanation: The backup volume on the indicated device will be clipped to the indicated generated or specified volser.

User response: No action is required.

**CKZ06039I BACKUP VOLUME ON DEVN: dddd
IS ALREADY ONLINE WITH
SPECIFIED VOLSER: vvvvvv**

Explanation: The backup volume on the indicated device is already with the specified volser.

User response: No action is required.

**CKZ06040E CKZ00900 UNEXPECTED RESULTS;
error text**

Explanation: An unexpected condition occurred calling program CKZ00900. Error text has a description of the problem. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ06041E INTERNAL ERROR; LOC=IIIII

Explanation: An internal error has occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ06042I VOLUME PAIRS BEING USED: list of
pairs**

Explanation: The listed pairs were found in the backinfo data set.

User response: No action is required.

| **CKZ06043E There is a blank storage group name for
| a volume in the backinfo file.**

| **Explanation:** A blank storage group name was found
| for a volume in the backinfo data set. The
| USERSGDEFS-DDN keyword has been specified, and
| the backinfo data set must have storage group names
| for all the volumes. Processing terminates.

| **User response:** Check that a correct backinfo data set
| is being used and that the correct version of the
| program that creates the backinfo data set is being
| used. If unable to resolve the problem, contact IBM
| Software Support.

**CKZ06050E ERROR IN PARAMETERS FOR
keyword**

Explanation: The parameters for the indicated keyword were incorrect. Processing terminates.

User response: Check the keyword parameters. Mutually exclusive keywords may have been used.

**CKZ06051E REQUIRED KEYWORD MISSING:
keyword**

Explanation: A keyword required for processing has been omitted. Processing terminates.

User response: Specify the required keyword.

**CKZ06052E THE SAME DDNAME HAS BEEN
SPECIFIED FOR MULTIPLE
KEYWORDS: ddname**

Explanation: The indicated ddname has been specified in multiple keywords. The specified ddnames must all be different. Processing terminates.

User response: Specify different ddnames in the keywords.

**CKZ06053E KEYWORD: keyword MAXIMUM
LENGTH: nnn EXCEEDED**

Explanation: The operand entered for a keyword exceeded the maximum length allowed for the operand. nnn is the maximum allowed length for the keyword. Processing terminates.

User response: Correct the length of the keyword's operand.

**CKZ06054E KEYWORD HAS MORE THAN 1
OPERAND; ONLY 1 ALLOWED:
keyword**

Explanation: Multiple operands were detected for a keyword; only one operand is permitted. Processing terminates.

User response: Correct the keyword to use one operand.

**CKZ06056E NOTHING SPECIFIED FOR
KEYWORD: keyword**

Explanation: A keyword was entered without an appropriate operand. Processing terminates.

User response: Specify an appropriate operand for the keyword.

**CKZ06057E DUPLICATE FOUND; KEYWORD:
keyword ENTRY: entry**

Explanation: The indicated entry for the keyword was previously specified. Processing terminates.

User response: Remove the duplicate entry.

**CKZ06058E INVALID VALUE IN KEYWORD:
keyword VALUE: value error text**

Explanation: The value in the keyword is invalid. 'error text' indicates the problem detected with the value. Processing terminates.

User response: Correct the value specified in the keyword.

**CKZ06060E UCBLOOK ERROR; RETURN
CODE=nn REASON CODE=nn
LOC=lllll**

Explanation: An error occurred during UCBLOOK processing. Processing terminates.

User response: Contact IBM Software Support. Have available the listing containing this message.

**CKZ06064E INVALID VOLSER: volser IN
KEYWORD: keyword**

Explanation: The volume serial number specified is invalid. Processing terminates.

User response: Correct the volser specification.

**CKZ06068E UNMATCHED ENTRIES IN
KEYWORD: keyword**

Explanation: For USERCATALOGS, there must be a source BCS followed by a target BCS. An uneven number of BCSs was specified. For VOLSER-RENAME-MASKS, there must be a source mask followed by a backup mask. An uneven number of masks was specified. Processing terminates.

User response: Correct the keyword specification.

**CKZ06081I DSNS FOR KEYWORD: keyword list of
dsns**

Explanation: Parsing found the listed dsns for the keyword.

User response: No action is required.

**CKZ06082I VOLSER RENAME MASK PAIRS FOR
KEYWORD: keyword list of mask pairs**

Explanation: Parsing found the listed mask pairs for the keyword.

User response: No action is required.

CKZ06083W VOLSER-RENAME-MASKS SPECIFIED BUT WILL NOT BE USED DUE TO CLIP-IF-OFFLINE(N)

Explanation: The VOLSER-RENAME-MASKS keyword has been specified, but CLIP-IF-OFFLINE(N) has also been specified or defaulted to. The volser rename masks will not be used.

User response: None, or remove the VOLSER-RENAME-MASKS keyword, or use CLIP-IF-OFFLINE(Y).

CKZ06086I VALIDATING KEYWORD: keyword

Explanation: Parsing is checking the indicated keyword in the command.

User response: No action is required.

CKZ07000I RESTORE-FROM-DUMPTAPES started - program rev=revision

Explanation: This message indicates the beginning of processing for RESTORE-FROM-DUMPTAPES, and includes the program revision level.

User response: No action is required.

CKZ07002I RESTORE-FROM-DUMPTAPES completed; return code=return_code

Explanation: This message indicates the end of processing for RESTORE-FROM-DUMPTAPES and includes the return code.

User response: No action is required.

CKZ07004E ddname missing: ddname

Explanation: An open failed for the ddname that is listed in the message, or the ddname is missing.

User response: If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07004E ddname missing: ddname

Explanation: An open failed for the ddname that is listed in the message, or the ddname is missing.

User response: If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07005E Open failed for ddname: ddname

Explanation: An open failed for the ddname that is listed in the message.

User response: If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07008E Unable to load program: load_module

Explanation: The load macro failed for the load module that is listed in the message.

User response: Ensure that the STEPLIB contains the load module, or that the load module is available via normal system search. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07010W Error calling CKZ01HEX; function=function R15=return_code

Explanation: An error occurred when attempting to format output into hexadecimal for display. The error occurred in function *function* and with return code *return_code*.

User response: If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07011E Error accessing STATUS file; loc=location

Explanation: A STATUS file open failed at the given location (for internal use only).

User response: Contact IBM Software Support.

CKZ07012E Duplicate STATUS file entry; loc=location

Explanation: An attempt to insert the initial control record failed because a duplicate record was found.

User response: If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07013E STATUS file control record not found. message_text

Explanation: If the RERUN keyword was specified, the control record must be present during initialization. If the RERUN keyword was not specified, an attempt to update the control record failed.

User response: Contact IBM Software Support.

CKZ07014E STATUS file control record version record_version is not current with program version program_version

Explanation: An older STATUS file format was specified with a newer version of RESTORE-FROM-DUMPTAPES.

User response: If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07015E STATUS file control record mismatch - type=*type* prevents use of RERUN

Explanation: The control record in the STATUS file does not match the current RESTORE-FROM-DUMPTAPES command input.

User response: If RERUN is required, it is possible that the wrong STATUS file is being used to RERUN the job; locate and specify the correct STATUS file. If RERUN is not required, remove the RERUN keyword, delete the STATUS file, and create a new STATUS file in the RESTORE-FROM-DUMPTAPES JCL. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07016E STATUS file control record shows Status-Complete - prevents use of RERUN

Explanation: The RERUN keyword was used, but the control record in the STATUS file shows that the clone finished with RC=0 or 4. Therefore, the clone cannot be rerun.

User response: Remove the RERUN keyword, delete the STATUS file, and create a new STATUS file in the RESTORE-FROM-DUMPTAPES JCL. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07017E STATUS file restore records not found - prevents use of RERUN

Explanation: There is no rerun information other than the control record.

User response: Remove the RERUN keyword, delete the STATUS file, and create a new STATUS file in the RESTORE-FROM-DUMPTAPES JCL. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07018E Keyword *keyword* not specified and default ddname *ddname* not supplied

Explanation: If the default ddnames for BACKINFO, VOLPAIRS, and UCATS are not supplied in the JCL, then for each of those not supplied, the appropriate BACKINFO-DDN, VOLPAIRS-DDN, or USERCATALOGS-DDN keyword must be specified.

User response: Use the default ddnames, or supply the correct DDN keyword. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07019E ddname *ddname* is not LRECL=80

Explanation: The backinfo data set (or the BACKINFO-DDN ddname) must be LRECL=80.

User response: Rerun the DB2GETBACKINFO

command to generate a new backinfo data set, using LRECL=80 on the DD statement. If you are unable to resolve this error, contact IBM Software Support.

CKZ07020W Record type *type* in *ddname* file is invalid type - ignored

Explanation: An invalid record type was encountered in the backinfo data set.

User response: If possible, remove the invalid record type from the backinfo data set. However, this warning can be ignored and processing continues. If you are unable to determine the cause of the warning, contact IBM Software Support.

CKZ07021E Expected continuation record not found: *record*

Explanation: The previous backinfo record indicated that a continuation record would follow, but no continuation record was found.

User response: Rerun the DB2GETBACKINFO command to generate a new backinfo data set. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07022E Unexpected continuation record: *record*

Explanation: An unexpected continuation record was encountered. The backinfo data set might be corrupted.

User response: Rerun the DB2GETBACKINFO command to generate a new backinfo data set. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07023E No DUMPTP records found in BACKINFO file

Explanation: No restores can be performed because no dump tapes are defined in the backinfo data set.

User response: Rerun the DB2GETBACKINFO command to generate a new backinfo data set. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07024W Volume *volume* currently is offline - will be skipped

Explanation: Target volume *volume* is offline and will not be used as a restore target.

User response: Determine if the volume in question should be online or if too many offline volumes prevent the job from continuing. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07025I Volume *volume* currently online - unit
address *unit_address*

Explanation: The volume that is listed in the message is currently online. The volume address is listed in the message.

User response: No action is required.

CKZ07042E ERROR CKZ0900 : *function*:
additional_message_text

Explanation: A parsing error was encountered on a keyword given by the *function*. If available, additional information may be provided in the *additional_message_text* field.

User response: Contact IBM Software Support.

CKZ07048I Options in effect for this execution:

Explanation: This message provides a summary of the runtime options after pairing volumes, resolving storage group names, and other input-related tasks.

User response: No action is required.

CKZ07049E User catalog *user_catalog* not found in
backinfo_ddname file

Explanation: The user catalog listed in the message was not found in the backinfo data set (or the *ddname* given in the BACKINFO-DDN keyword).

User response: Remove the user catalog from the USERCATALOGS keyword, or rerun the DB2GETBACKINFO command to generate a new backinfo data set. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07050E Keyword USERCATALOGS specified,
also requires keyword
USERCATALOGS-DDN (not found)

Explanation: The USERCATALOGS keyword was supplied, but either the UCATS *ddname* was not provided in the JCL, or the USERCATALOGS-DDN keyword was not supplied. One or the other is required with the USERCATALOGS keyword.

User response: Add the UCATS DD to the JCL, or a different DD specified by USERCATALOGS-DDN, and resubmit. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07051E ERROR - Keyword: *keyword* is required

Explanation: The keyword listed in the message is required, but it was not supplied.

User response: Add the missing keyword and resubmit.

CKZ07052E ERROR - Keyword: *keyword* has odd
number of values, must be even number
of paired values

Explanation: The keyword listed in the message has an odd number of values. The number of values must be even.

User response: Correct the input and resubmit.

CKZ07053E ERROR - Keyword: *keyword error_text*
additional_message_text

Explanation: The keyword that is listed in the message has an error. The error is described in the message text. Possible values for *error_text* are:

- value too long, max length = *xx*
- value must be numeric
- value must be 1-16 inclusive

More information might be provided in the *additional_message_text* field.

User response: Correct the input and resubmit.

CKZ07055E Keyword USERCATALOGS specified
but no UCAT records found in *ddname*
file

Explanation: The USERCATALOGS keyword requires the same user catalogs to be defined in the backinfo data set (identified in the message by *ddname*).

User response: Rerun the DB2GETBACKINFO command with the USERCATALOGS keyword and the correct user catalogs to regenerate the backinfo data set. Use the new backinfo data set and rerun RESTORE-FROM-DUMPTAPES. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07058E Invalid value in keyword: *keyword* *value*:
value

Explanation: The keyword and its value that are listed in message are invalid.

User response: Refer to the documentation for valid values for this command and keyword. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07060E UCBLOOK error; return
code=return_code *reason code=reason_code*
LOC=location

Explanation: The UCBLOOK macro failed with the given return code and reason code and at the given location (for internal use only).

User response: If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07061E CKZ01SMF error; return code=return_code loc =location additional_message_text

Explanation: Storage group lookup failed with the give return code that is listed in the message, at the given location (for internal use only). More information might be provided in the *additional_message_text* field; also refer to accompanying message CKZ07062E.

User response: Refer to message CKZ07062E for the name of the storage group that returned the error. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07062E No volumes returned by SSI for storage group: storage_group

Explanation: This message is used with CKZ07061E, and indicates that storage group lookup failed for the storage group that is listed in the message. No volumes were found.

User response: Verify the storage group name. If necessary, consult your DASD administrators. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07063E Maximum volumes reached = 9999

Explanation: The maximum number of volumes for one command was exceeded.

User response: Reduce the number of volumes for one command by dividing the backinfo data set into separate files and using multiple RESTORE-FROM-DUMPTAPES commands. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07067E No volumes in storage group(s)

Explanation: None of the specified target storage groups have volumes assigned to them.

User response: Correct the target storage group names, or add volumes to the target storage groups. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07071E More source volsers number_of_source_volsers than online target volsers number_of_target_volsers

Explanation: The number of source volumes exceeds the number of online target volumes.

User response: Add more target volumes or storage groups to the target list, or bring more target volumes online. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07072E More source volsers number_of_source_volsers than target volsers number_of_target_volsers

Explanation: The number of source volumes exceeds the number of target volumes.

User response: Add more target volumes or storage groups to the target list. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07073E Volume matching error for target volume target_volsers

Explanation: The target volume listed in the message could not be matched with any source volume or dumptape. Possible reasons include: the volume is part of a storage group, but the source storage group that it is matched with is empty; or all remaining source volumes are too large to match with this target volume.

User response: Investigate the potential cause based on the suggestions above. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07076I Volume target_volsers paired with source volume source_volsers

Explanation: This informational message lists the target volume and source volume pair.

User response: No action is required.

CKZ07077E User catalog lookup error error_text

Explanation: The catalog search interface failed for the reason given in *error_text*.

User response: If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07078E User catalog user_catalog resides on target volume target_volume

Explanation: Target user catalogs may not reside on target volumes.

User response: Move the user catalog to a different volume before rerunning the RESTORE-FROM-DUMPTAPES command. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07087E Masking characters not allowed with keyword keyword

Explanation: Masking characters are not supported with the keyword that is listed in the message.

User response: Use specific values for this keyword. If you are unable to determine the cause of this error, contact IBM Software Support.

| **CKZ07089E** **Limit of 32,000 wildcard volumes exceeded**

|

| **Explanation:** No single wildcard mask for volume names can exceed 32,000 volumes. The overall limit for all volumes derived from all wildcard masks is also 32,000.

| **User response:** Use wildcard masks that return a smaller number of volumes. If you are unable to determine the cause of this error, contact IBM Software Support.

| **CKZ07090I** **Volume *volser* included via volume mask *volume_mask***

|

| **Explanation:** The *volser* that is listed in the message is included because it matches the given volume mask.

| **User response:** No action is required.

| **CKZ07102I** **End of CKZ00071 processing – max rc return_code**

|

| **Explanation:** Program CKZ00071 is complete and returned the return code that is listed in the message.

| **User response:** No action is required.

| **CKZ07104E** **LOAD failed for module: *module_name***

|

| **Explanation:** An attempt to load the program that is listed in the message text failed.

| **User response:** Ensure that the DB2 Cloning Tool runtime libraries are included in the STEPLIB concatenation. If you are unable to determine the cause of this error, contact IBM Software Support.

| **CKZ07105E** **Allocation failed for ddname: *ddname***

|

| **Explanation:** An attempt to allocate the file that is listed in the message text failed.

| **User response:** Ensure that the DB2 Cloning Tool runtime libraries are included in the STEPLIB concatenation. If you are unable to determine the cause of this error, contact IBM Software Support.

| **CKZ07106W** **Deallocation failed for ddname: *ddname***

|

| **Explanation:** An attempt to deallocate the file that is listed in the message text failed.

| **User response:** This warning might not be a problem. If you are unable to determine the cause of this message, contact IBM Software Support.

| **CKZ07107E** **ATTACHX failed RC *return_code* additional_message_text**

|

| **Explanation:** An attempt to attach a subtask failed with the return code that is listed in the message.

| **User response:** Ensure that the DB2 Cloning Tool runtime libraries are included in the STEPLIB concatenation. If you are unable to determine the cause of this error, contact IBM Software Support.

| **CKZ07111E** **Error accessing STATUS file; loc=*location***

|

| **Explanation:** An attempt to insert, read with update, or rewrite a record to the STATUS file failed.

| **User response:** Save the job listing and the STATUS file and contact IBM Software Support.

| **CKZ07112E** **Duplicate STATUS file entry; loc=*location***

|

| **Explanation:** An attempt to insert a record into the STATUS file failed.

| **User response:** Save the job listing and the STATUS file and contact IBM Software Support.

| **CKZ07114E** **STATUS file restore record not found**

|

| **Explanation:** An attempt to read-with-update a record in the STATUS file failed because the record was not found.

| **User response:** Save the job listing and the STATUS file and contact IBM Software Support.

| **CKZ07118E** **ERROR: failure *condition_code* at *module_name* + *offset* - ***** TERMINATING *******

|

| **Explanation:** An unrecoverable program error occurred. The completion code, module name and offset are listed in the message text.

| **User response:** Save the job listing and the STATUS file and contact IBM Software Support.

| **CKZ07119E** **Unable to establish ESTATEX; R15=*return_code***

|

| **Explanation:** An attempt to establish error recovery via the ESTAEX macro failed with the return code listed in the message text.

| **User response:** Save the job listing and the STATUS file and contact IBM Software Support.

CKZ07120W **Skipping clip / vary for volume** *volser*
due to prior restore task error

Explanation: Failure of the restore task prevents further processing of the volume that is listed in the message.

User response: Resolve the cause of the restore task error before proceeding. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07121E **Fatal error with clip / vary on volume**
volser - exiting

Explanation: An essential part of the post-restore processing for the volume that is listed in the message failed.

User response: Review other associated messages to determine and fix the error. If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07125E <<>> **Found** *number* **offline target**
volumes - TERMINATING

Explanation: There are not enough target volumes available online to continue. The number of offline volumes from the pool of target volumes is listed in the message. The command terminates with a return code of 8.

User response: Specify a different set of target volumes, or bring more target volumes online, or both.

CKZ07130I **Restore task completed: target** *volser* **tape**
tape_volume **file** *file_sequence_number*
return code *return_code*

Explanation: This message indicates that a single restore for the target volume and tape volume listed in the message completed with return code of *return_code*.

User response: If the return code is nonzero, review other applicable messages. Otherwise, no action is required.

CKZ07131E **Remaining restores incomplete:**

Explanation: One or more errors occurred that leave some restores incomplete. Details are provided in the accompanying CKZ07132I messages.

User response: Refer to other messages that are related to this error in the job log or DB2 Cloning Tool message file. Rerun the job with appropriate changes. If you are unable to resolve this error, have the job listing and associated files available, and contact IBM Software Support.

CKZ07132I *target_volser* **requiring tape** *tape_volume*
additional_information

Explanation: This message accompanies message CKZ07131E and provides details for a target volume that did not get restored and the tape volumes that it requires. Additional information may be provided in the message.

User response: No action is required.

CKZ07140E **Unexpected condition:** *error_text*

Explanation: An unexpected condition was encountered.

User response: Have the job listing and associated files available, and contact IBM Software Support.

CKZ07160E **UCBLOOK error: return**
code=return_code **reason code**=*reason_code*
LOC=*location*

Explanation: The UCBLOOK macro failed. The return and reason codes are listed in the message, as well as the internal location given by *location*.

User response: Have the job listing and associated files available, and contact IBM Software Support.

CKZ07201I **Restore started - program** *rev=revision* [
**** SIMULATION ****]

Explanation: The restore driver for one target volume began execution. The program revision is listed in the message.

User response: No action is required.

CKZ07202I **Restore completed; return**
code=return_code

Explanation: The restore driver for one target volume completed with the return code that is listed in the message.

User response: No action is required.

CKZ07203I **DDname**=*ddname* **allocated for**
dsn=data_set_name

Explanation: The ddname that is listed in the message has been allocated for the data set name *data_set_name*. For tape allocations, *data_set_name* display as ****TAPE****, followed by the tape volume and file sequence number.

User response: No action is required.

CKZ07204E Open failed for *ddname=ddname*

Explanation: An attempt to open the data set with *ddname ddname* failed.

User response: If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07205E Allocation failed for *dsn=data_set_name*

Explanation: An attempt to allocate the data set that is listed in the message failed.

User response: If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07206W Deallocation failed for *ddname: ddname*

Explanation: An attempt to deallocate the data set for the *ddname* that is listed in the message failed.

User response: This message is a warning and may not affect the restore process. If you are unable to determine the cause of this warning, contact IBM Software Support.

CKZ07214I Tape *tape_volume, fileseq*
file_sequence_number, dsn data_set_name **
SIMULATION **

Explanation: This message displays in simulation mode and lists the tape volume, file sequence number, and data set name that would be allocated for this restore during a non-simulation run.

User response: No action is required.

CKZ07221E ADRDSSU restore failed; R15:
return_code

Explanation: The restore program ADRDSSU failed with the return code that is listed in the message.

User response: If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07301I Clip target started - program *rev=*
revision

Explanation: The process of clipping a target volume started. The revision number of the program is shown in the message.

User response: No action is required.

CKZ07302I Clip target volume *volume devn*
device_number completed; return code=
return_code

Explanation: The process of clipping a target volume completed with the listed *return_code*. The information for the volume that was clipped is shown in the message text.

User response: No action is required.

CKZ07303I *ddname ddname* allocated for *dsn*
data_set_name

Explanation: The data set that is listed in the message was successfully allocated with *ddname ddname*.

User response: No action is required.

CKZ07304E Open failed for *ddname ddname*

Explanation: The *ddname* that is listed in the message cannot be opened.

User response: If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07305E Allocation failed for *dsn data_set_name*

Explanation: The data set that is listed in the message could not be allocated.

User response: If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07306W Deallocation failed for *ddname ddname*

Explanation: The *ddname* that is listed in the message could not be deallocated.

User response: If you are unable to determine the cause of this warning, contact IBM Software Support.

CKZ07335E ICKDSF command failed for target
volume *volume*

Explanation: An attempt to relabel the target volume that is listed in the message failed.

User response: If you are unable to determine the cause of this warning, contact IBM Software Support.

CKZ07336E IEEVARYD vary on/offline failed for
target volume *volume*; invalid parameters

Explanation: An attempt to vary the target volume that is listed in the message online or offline failed due to invalid parameters.

User response: Contact IBM Software Support.

CKZ07337W IEEVARYD vary on/offline failed for
target volume *volume* device *device* return
code *return_code* reason code *reason_code*

Explanation: An attempt to vary the target volume that is listed in the message online or offline failed due to invalid parameters. Details about the target volume are provided in the message text.

User response: Contact IBM Software Support.

CKZ07338W IEEVARYD vary on/offline failed for target volume *volume* device *device* R15 *return_code*

Explanation: An attempt to vary the target volume that is listed in the message online or offline failed. The device information and return code are provided in the message.

User response: If you are unable to determine the cause of this warning, contact IBM Software Support.

CKZ07339E IEEVARYD vary offline failed for target volume *volume* device *device* – retry loop exhausted

Explanation: An attempt to vary the target volume that is listed in the message offline failed until the retry loop is exhausted.

User response: If you are unable to determine the cause of this error, contact IBM Software Support.

CKZ07340I Device number *device* successfully changed to volume serial *volser*

Explanation: The device was successfully relabeled with the volume serial that is listed in the message.

User response: No action is required.

CKZ07341I Volume serial *volser* device number *device* is now online

Explanation: The volume that is listed in the message is now online.

User response: No action is required.

CKZ07348I Volume serial *volser_1* device number *device* not clipped to volume serial *volser_2* - ** SIMULATION **

Explanation: *volser_1* at the device number *device* was not relabeled to volume *volser_2* because the command was run in SIMULATION mode.

User response: No action is required.

CKZ10001I hh:mm:ss RENAME PROCESS STARTED - PROGRAM REV=rrr | hh:mm:ss RENAME PROCESS COMPLETED; RETURN CODE=nnn

Explanation: RENAME command processing message.

User response: No action is required.

CKZ10003I DDNAME=ddname ALLOCATED FOR DSN=datasetname

Explanation: 'ddname' has been dynamically allocated for the indicated data set.

User response: No action is required.

CKZ10004E DDNAME MISSING: ddname | OPEN FAILED FOR DDNAME: ddname

Explanation: 'ddname' was specified for DB2 Cloning Tool to use. Processing terminates.

User response: Either correct the ddname specified, or, add the appropriate ddname to the job's JCL.

CKZ10005E ALLOCATION FAILED FOR DSN: datasetname | ALLOCATION FAILED FOR DDNAME: ddname

Explanation: Dynamic allocation for a data set or ddname failed. The associated z/OS messages are displayed. Processing terminates.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ10005W DEALLOCATION FAILED FOR DDNAME: ddname

Explanation: Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ10006E ERROR CALLING CKZ01VV1 tttttt FUNCTION: function R15=nnnn R0=nnnnnnnn LOC=lllll

Explanation: A problem occurred using a dataspace. tttttt is the name of the internal table. lllll is the location where the error occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

CKZ10007W ERROR CALLING CKZ01HEX; FUNCTION: function R15=nnnn

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

**CKZ10008E UNABLE TO LOAD PROGRAM:
program name**

Explanation: The indicated program name was not found. Processing terminates.

User response: Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

**CKZ10009E ERROR ACCESSING JOURNAL FILE;
LOC=IIIII**

Explanation: A VSAM error occurred accessing the journal file. Processing terminates.

User response: See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ10010E DUPLICATE JOURNAL ENTRY;
LOC=IIIII**

Explanation: A duplicate record was detected. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ10011E JOURNAL CONTROL RECORD NOT
FOUND | JOURNAL CONTROL
RECORD IS WRONG VERSION |
JOURNAL USER CATALOG
RECORD(S) NOT FOUND | JOURNAL
VOLUME PAIR RECORD(S) NOT
FOUND | JOURNAL EXCLUDE
RENAME MASK RECORD(S) NOT
FOUND**

Explanation: An expected record was not found in the DB2 Cloning Tool journal file. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ10013E RECORD COUNT IS ZERO; LOC=IIIII
| COUNT MISMATCH; RECORDS
READ=rrrr CONTROL RECORD
COUNT=cccc**

Explanation: There was a problem with the journal records needed to initiate the command. For the first format, the journal control record indicates no entries were added. For the second format, the number of records read from the journal, rrrr, is not the same as the number indicated in the journal control record, cccc. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ10015E THE COPY PROCESS DID NOT
COMPLETE SUCCESSFULLY**

Explanation: The journal indicates that the COPY command did not complete successfully. Processing terminates.

User response: Check that the COPY command has completed successfully before initiating the RENAME command.

**CKZ10016I COPY WAS A SIMULATION; RENAME
CHANGED TO SIMULATION**

Explanation: The journal indicates that the COPY command was a simulation. Processing continues, but, the RENAME will be run as a simulation.

User response: No action is required.

**CKZ10017E THE DDNAME IS EMPTY OR HAS
BEEN DUMMIED, DDNAME: ddn**

Explanation: No records were read from the ddname specified for a keyword. Processing terminates.

User response: Check that the DD has not been specified as 'DD DUMMY' or 'DD DSN=NULLFILE'. Check that the DSN specified in the ddn has been created successfully.

**CKZ10019E THE keyword DOES NOT HAVE A
LRECL OF 80, DDNAME: ddn**

Explanation: The data set allocated to the ddname for the identified keyword does not have a LRECL of 80. The LRECL of this data set must be 80. Processing terminates.

User response: Change the data set allocated to the ddname to have a LRECL of 80.

**CKZ10020E SOURCE CATALOG BACKUP HAS
NOT BEEN DONE**

Explanation: The source catalogs have not been backed up. Processing terminates.

User response: Run UCATOPTIONS BACKUP to backup the source catalogs.

CKZ10040E *multiple possible messages; see Explanation*

Explanation:

ERROR ATTACHING DRIVER, PGM=program name
The indicated program name was not found.
Processing terminates.

UNEXPECTED CONDITION, TASK NOT POSTED
The program returned from a 'WAIT', but, had
not been posted. Processing terminates.

User response: For ERROR ATTACHING DRIVER message,

CKZ10041E • CKZ10056E

check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

For UNEXPECTED CONDITION message, contact IBM Software Support. Have available the listing that contains this message.

CKZ10041E CKZ00900 UNEXPECTED RESULTS; error text

Explanation: An unexpected condition occurred calling program CKZ00900. 'error text' has a description of the problem. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ10042E RERUN NOT SPECIFIED AND PREVIOUS RUN WAS NOT A SIMULATION

Explanation: A RENAME was attempted without the RERUN parameter, but the journal indicates that a non-simulation RENAME has already been attempted.

User response: If 'SAFE' mode was used for the first RENAME, specify 'RERUN' for the RENAME command.

CKZ10043E VOLBKUP DSNNAME MISMATCH: CURRENT DSN=current dsn in command PREVIOUS DSN=previous dsn used

Explanation: A RENAME with RERUN is pointing to a different data set for the volume backup file.

User response: Correct the data set name for the volume backup file in the JCL.

CKZ10044E RERUN WAS SPECIFIED AND PREVIOUS RUN WAS NOT SAFE

Explanation: A RENAME RERUN was attempted, but the journal indicates that a previous RENAME did not include the SAFE parameter.

User response: The COPY command will need to be run before initiating the RENAME command without the RERUN parameter.

CKZ10048I OPTIONS IN EFFECT FOR THIS EXECUTION: merged options

Explanation: Informational message indicating how RENAME will handle the options. The displayed options are derived from the CKZINI and any overriding specifications in the command input.

User response: No action is required.

CKZ10050E ERROR IN PARAMETERS FOR keyword

Explanation: The parameters for the indicated keyword were incorrect. Processing terminates.

User response: Check the keyword parameters. Mutually exclusive keywords may have been used.

CKZ10051E REQUIRED KEYWORD MISSING: keyword

Explanation: A keyword required for processing has been omitted. Processing terminates.

User response: Specify the required keyword.

CKZ10052E REQUIRED INI SECTION/TOKEN MISSING: SECTION=section TOKEN=token | REQUIRED INI VALUE MISSING FOR SECTION=section TOKEN=token | INVALID INI VALUE FOR SECTION=section TOKEN=token

Explanation: An error occurred validating the CKZINI member options. Processing terminates.

User response: Correct the CKZINI member.

CKZ10053E KEYWORD: keyword MAXIMUM LENGTH: nnn EXCEEDED

Explanation: The operand entered for a keyword exceeded the maximum length allowed for the operand. nnn is the maximum allowed length for the keyword. Processing terminates.

User response: Correct the length of the keyword's operand.

CKZ10054E KEYWORD HAS MORE THAN 1 OPERAND; ONLY 1 ALLOWED: keyword

Explanation: Multiple operands were detected for a keyword; only one operand is permitted. Processing terminates.

User response: Correct the keyword to use one operand.

CKZ10056E NOTHING SPECIFIED FOR KEYWORD: keyword

Explanation: A keyword was entered without an appropriate operand. Processing terminates.

User response: Specify an appropriate operand for the keyword.

CKZ10057E DUPLICATE FOUND; KEYWORD:
keyword ENTRY: entry

Explanation: The indicated 'entry' for the keyword was previously specified. Processing terminates.

User response: Remove the duplicate entry.

CKZ10058E INVALID VALUE IN KEYWORD:
keyword VALUE: value error text

Explanation: The value in the keyword is invalid. 'error text' indicates the problem detected with the value. Processing terminates.

User response: Correct the value specified in the keyword.

CKZ10061E CKZ01SMF ERROR; RETURN
CODE=nnnn LOC: llllll INVALID
smstypeCLAS NAME: class entry

Explanation: An error occurred using CKZ01SMF to verify the SMS class specified. llllll is the internal location where the error occurred. smstype indicates DATA, MGMT, or, STOR. Processing terminates.

User response: Correct the indicated SMS class entry.

CKZ10068E UNPAIRED ENTRIES IN KEYWORD:
keyword

Explanation: The keyword requires pairs of entries. An odd number of entries was found. Processing terminates.

User response: Correct the keyword specification.

CKZ10069E SMF IS NOT RECORDING THE
SPECIFIED RECORD TYPE: nnn

Explanation: The SMF audit log has been requested but SMF is not recording the specified record type. Processing terminates.

User response: Correct the keyword specification to use a record type that SMF is recording or have SMF record the specified record type.

CKZ10085I DSNS FOR KEYWORD: keyword
PROCESSING SEQUENCE list of dsns
number

Explanation: Parsing found the listed dsns for the keyword. The processing sequence number shows the order that the dsns were entered and will be the order used during processing.

User response: No action is required.

CKZ10086I VALIDATING KEYWORD: keyword

Explanation: Parsing is checking the indicated keyword indicated in the command.

User response: No action is required.

CKZ11001I hh:mm:ss VOLUME UPDATES
STARTED - PROGRAM REV=rrr (**
SIMULATION **) | hh:mm:ss VOLUME
UPDATES COMPLETED; RETURN
CODE=nnn

Explanation: VOLUME UPDATES processing message.

User response: No action is required.

CKZ11004E OPEN FAILED FOR DDNAME=ddname

Explanation: 'ddname' was allocated for DB2 Cloning Tool to use, but, the open for the file failed. Processing terminates.

User response: If unable to determine the reason the open failed, contact IBM Software Support. Have available the listing that contains this message.

CKZ11006E ERROR CALLING CKZ01VV1 tttttt
FUNCTION: function R15=nnnn
R0=nnnnnnnn LOC=lllll

Explanation: A problem occurred using a dataspace. tttttt is the name of the internal table. llll is the location where the error occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

CKZ11007W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

CKZ11009E ERROR ACCESSING JOURNAL FILE;
LOC=lllll

Explanation: A VSAM error occurred accessing the journal file. Processing terminates.

User response: See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ11011E JOURNAL VOLUME PAIR RECORD NOT FOUND FOR source volume/target volume

Explanation: An expected record was not found in the DB2 Cloning Tool journal file. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ11024E VOLUME BACKUP FAILED FOR volser. SEE SYSOUT DRSTATS FOR MORE INFORMATION

Explanation: An error occurred during SAFE mode attempting to back up a volume. Processing terminates.

User response: Check the DRSTATS for the error that occurred. If unable to correct the problem, contact IBM Software Support.

CKZ11030I VOLUME CONVERSION STARTED FOR source volume/target volume | VOLUME CONVERSION COMPLETED FOR source volume/target volume; RETURN CODE=nn DATA SETS=nnnnnn | VOLUME CONVERSION FAILED FOR source volume/target volume; RETURN CODE=nn DATA SETS=nnnnnn

Explanation: Volume processing information message. DATA SETS=nnnnnn indicates the number of format 1 dscb's renamed on the target volume.

User response: None, unless the message includes the word FAILED or the RETURN CODE= is greater than four (4). If so, check the sysout for that target volume for the error messages.

CKZ11031I VOLUME CONVERSION TOTAL DATA SETS: nnnnnnnn

Explanation: Volume processing information message. The number is the total number of format 1 dscb's renamed on all target volumes.

User response: No action is required.

CKZ11032I VOLUME BACKUP COMPLETED FOR target volume

Explanation: Volume processing information message. During 'SAFE' mode, the target volume's VTOC, VTOCIX, and VVDS are backed up before any changes occur.

User response: No action is required.

CKZ11033W VOLUME CONVERSION BYPASSED FOR source volume/target volume DUE TO PRIOR ERRORS

Explanation: Volume processing has not been done for the target volume due to errors that happened while processing another volume. The other volumes with errors can be determined from the prior messages: CKZ11030I VOLUME CONVERSION FAILED FOR srcvol/tgtvol

User response: Correct the cause for the errors that happened to the other volumes and do a RENAME RERUN if possible. If RENAME RERUN is not possible the COPY will have to be redone.

CKZ11034I TOTAL BYTES WRITTEN TO VOLBKUP DDNAME=ddname; nnnnnnnn

Explanation: Volume processing information message. This shows the number of bytes that were written to the VOLBKUP file.

User response: No action is required.

CKZ11035I TARGET VOLUME volser IS AN EXTENDED ADDRESS VOLUME; ICKDSF WILL BE USED TO REBUILD THE VTOCIX

Explanation: The identified volume is an Extended Address Volume and VTOCIX_REBUILDER = MSC has been specified in the INI. For Extended Address Volumes ICKDSF will always be used to rebuild the VTOCIX.

User response: No action is required.

CKZ11040E UNEXPECTED CONDITION; error text

Explanation: An unexpected condition occurred while dispatching a volume task or while waiting for the completion of a volume task. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ11041E TARGET VOLUME target volume IS NOT IN A PROPER STATE FOR RENAME

Explanation: The target volume is not in a proper state for RENAME processing. The volume may have been modified between COPY and RENAME which is not allowed. Processing terminates.

User response: The COPY will need to be run again to put the target volume into the proper state for RENAME processing.

CKZ11042W UNEXPECTED CONDITION; error text

Explanation: An unexpected condition occurred while dispatching a service task. Processing continues.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ11060E ERROR DURING function FOR
VOLSER=volume - RETURN
CODE=nnnn REASON CODE=nnnn**

Explanation: An error occurred using IOSCAPU, UCBLOOK, or UCBPIN. Processing terminates.

User response: A return code 4 from UCBLOOK may indicate the volser is offline. If unable to determine the cause of the error, contact IBM Software Support. Have available the listing containing this message.

**CKZ11089I NOT ALL DATASETS HAVE BEEN
RENAMED ON THE VOLUMES**

Explanation: An error occurred during volume rename processing that caused some data sets to not be renamed on a volume. Message CKZ11030I will indicate which volume pair had an error.

User response: Correct the cause of the error and rerun RENAME if possible.

**CKZ11101I hh:mm:ss VOLUME UPDATE STARTED
- PROGRAM REV=rrr (**
SIMULATION **) CKZ11101I hh:mm:ss
VOLUME UPDATE COMPLETED;
RETURN CODE=nnn F1DSCB
COUNT=nnnnnnnn**

Explanation: Individual VOLUME UPDATE processing message.

User response: No action is required.

**CKZ11102I RENAMING VTOC ENTRIES ON
VOLUME: target volume (**
SIMULATION **) | RENAMING
VTOCIX ENTRIES ON VOLUME: target
volume (** SIMULATION **)**

Explanation: Individual VOLUME UPDATE processing messages.

User response: No action is required.

CKZ11104E OPEN FAILED FOR DDNAME=ddname

Explanation: 'ddname' was allocated for DB2 Cloning Tool to use, but, the open for the file failed. Processing terminates.

User response: If unable to determine the reason the open failed, contact IBM Software Support. Have available the listing that contains this message.

**CKZ11105E ALLOCATION FAILED FOR DDNAME:
ddname**

Explanation: Dynamic allocation for a ddname failed. The associated z/OS messages are displayed. Processing terminates.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ11105W DEALLOCATION FAILED FOR
DDNAME: ddname**

Explanation: Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ11106E ERROR CALLING CKZ01VV1 tttttt
FUNCTION: function R15=nnnn
R0=nnnnnnnn LOC=lllll**

Explanation: A problem occurred using a dataspace. tttttt is the name of the internal table. lllll is the location where the error occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

**CKZ11107W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn**

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

**CKZ11109E ERROR ACCESSING JOURNAL FILE;
LOC=lllll**

Explanation: A VSAM error occurred accessing the journal file. Processing terminates.

User response: See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ11110W VOLSER IN FORMAT 1 NOT IN
VOLSER PAIRS: volser found DSN:
datasetname**

Explanation: The volume serial found in the format 1

CKZ11112W • CKZ11130E

dscb for the indicated data set is not in the volume pairs. Processing continues.

User response: This probably is a condition copied from the source volumes. If desired, correct the format 1 dscb on the source volume for the data set. (An incorrect volume serial in the format 1 dscb will not prevent accessing the data set.)

CKZ11112W DATA SET MATCHES NO RENAME MASK: datasetname (MATCHES EXCLUDE MASK)

Explanation: The indicated data set did not match any source rename mask. If the message includes (MATCHES EXCLUDE MASK) the data set matched an entry in the EXCLUDE-SRCNAME-MASKS keyword. Processing continues.

User response: No action is required.

CKZ11114E UNEXPECTED VALUE FOUND IN VVDS

Explanation: During rename of the SYS1.VVDS data set, something unexpected was found in the self-defining entry. The entry is printed. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ11115E VOLUME VOLSER NOT EQUAL TO NEW VOLSER: volser in label

Explanation: During volume processing, the volume serial in the label does not match the target volume. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ11117E RENAME WOULD CAUSE A DUPLICATE DSN IN VTOC: source dsn NEWNAME: target dsn

Explanation: The target dsn would result in a duplicate name in the VTOC. Processing terminates.

User response: Correct the RENAME-MASKS to prevent the creation of duplicate data set names.

CKZ11120E UNPOPULATED SECTION; VIR ENTRIES AVAILABLE

Explanation: An internal error occurred during VTOCIX processing.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ11121E EXPECTED VIXM ENTRY NOT FOUND

Explanation: An internal error occurred during VTOCIX processing.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ11122E EXPECTED VIB ENTRY NOT FOUND

Explanation: An internal error occurred during VTOCIX processing.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ11123E UNKNOWN RECORD TYPE FOUND IN VTOCIX

Explanation: An internal error occurred during VTOCIX processing.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ11124E UNABLE TO VALIDATE DCE FOR VOLUME: target volume

Explanation: The DCE fields for the target volume have not been set. Accessing the VTOC did not cause the DCE fields to be corrected.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ11130E RETURN CODE 8 SET FOR TEMPORARY DATA SET(S)

Explanation: A return code of 8 was requested for the indicated condition. Processing terminates.

User response: No action is required, unless a return code of 0 or 4 is desired for the condition. If a return code other than 8 is wanted, either change the CKZINI member, or, override the return code in the RENAME command.

CKZ11130E RETURN CODE 8 SET FOR TEMPORARY DATA SET(S)

Explanation: A return code of 8 was requested for the indicated condition. Processing terminates.

User response: No action is required, unless a return code of 0 or 4 is desired for the condition. If a return code other than 8 is wanted, either change the CKZINI member, or, override the return code in the RENAME command.

CKZ11131I RENAME DATA SETS:

Explanation: A RENAME SIMULATE was requested. The data set names that are listed will be renamed on the target volume.

User response: No action is required.

**CKZ11133E EXCP FAILED: function | SYNAD
TEXT: error text from SYNADAF**

Explanation: An EXCP request failed. SYNADAF was invoked to help diagnose the problem. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ11143E UNABLE TO RENAME DSN=source
datasetname | USING MASK=target
mask**

Explanation: The new name of a data set will exceed 44 characters. Processing terminates.

User response: Correct the RENAME-MASKS specification.

**CKZ11157E type VOLSER: volser IS EXTENDED
ADDRESS VOLUME**

Explanation: The indicated volser is an Extended Address Volume (EAV). Extended Address Volumes are not currently supported by DB2 Cloning Tool. Processing terminates.

User response: Exclude the volume from processing.

**CKZ11160E ERROR DURING function FOR
VOLSER=volume - RETURN
CODE=nnnn REASON CODE=nnnn**

Explanation: An error occurred using IOSCAPU, UCBLLOOK, or UCBPIN. Processing terminates.

User response: A return code 4 from UCBLLOOK may indicate the volser is offline. If unable to determine the cause of the error, contact IBM Software Support. Have available the listing containing this message.

**CKZ11161E ERROR CALLING SMFWTM;
R15=nnnn**

Explanation: An error occurred using the SMFWTM macro. Processing terminates.

User response: If unable to determine the cause of the error, contact IBM Software Support. Have available the listing containing this message.

**CKZ11201I hh:mm:ss VOLUME UPDATE STARTED
- PROGRAM REV=rrr (**
SIMULATION **) | hh:mm:ss VOLUME
UPDATE COMPLETED; RETURN
CODE=nnn F1DSCB
COUNT=nnnnnnnn F8DSCB
COUNT=nnnnnnnn**

Explanation: Individual VOLUME UPDATE processing message.

User response: No action is required.

**CKZ11202I RENAMING VTOC ENTRIES ON
VOLUME: target volume (**
SIMULATION **)**

Explanation: Individual VOLUME UPDATE processing message.

User response: No action is required.

**CKZ11203I DDNAME=ddname ALLOCATED FOR
DSN=datasetname**

Explanation: 'ddname' has been dynamically allocated for the indicated data set.

User response: No action is required.

CKZ11204E OPEN FAILED FOR DDNAME=ddname

Explanation: 'ddname' was allocated for DB2 Cloning Tool to use, but, the open for the file failed. Processing terminates.

User response: If unable to determine the reason the open failed, contact IBM Software Support. Have available the listing that contains this message.

**CKZ11205E ALLOCATION FAILED FOR DSN:
datasetname | ALLOCATION FAILED
FOR DDNAME: ddname**

Explanation: Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ11205W DEALLOCATION FAILED FOR
DDNAME: ddname**

Explanation: Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact

IBM Software Support. Have available the listing containing these messages.

**CKZ11206E ERROR CALLING CKZ01VV1 tttttt
FUNCTION: function R15=nnnn
R0=nnnnnnnn LOC=llll**

Explanation: A problem occurred using a dataspace. tttttt is the name of the internal table. llll is the location where the error occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

**CKZ11207W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn**

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

**CKZ11209E ERROR ACCESSING JOURNAL FILE;
LOC=llll**

Explanation: A VSAM error occurred accessing the journal file. Processing terminates.

User response: See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ11210W VOLSER IN FORMAT 1 NOT IN
VOLSER PAIRS: volser found DSN:
datasetname**

Explanation: The volume serial found in the format 1 dscb for the indicated data set is not in the volume pairs. Processing continues.

User response: This probably is a condition copied from the source volumes. If desired, correct the format 1 dscb on the source volume for the data set. (An incorrect volume serial in the format 1 dscb will not prevent accessing the data set.)

**CKZ11212W DATA SET MATCHES NO RENAME
MASK: datasetname (MATCHES
EXCLUDE MASK)**

Explanation: The indicated data set did not match any source rename mask. If the message includes (MATCHES EXCLUDE MASK) the data set matched an entry in the EXCLUDE-SRCNAME-MASKS keyword. Processing continues.

User response: No action is required.

**CKZ11214E UNEXPECTED VALUE FOUND IN
VVDS**

Explanation: During rename of the SYS1.VVDS data set, something unexpected was found in the self-defining entry. The entry is printed. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ11215E VOLUME VOLSER NOT EQUAL TO
NEW VOLSER: volser in label**

Explanation: During volume processing, the volume serial in the label does not match the target volume. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ11217E RENAME WOULD CAUSE A
DUPLICATE DSN IN VTOC: source dsn
NEWNAME: target dsn**

Explanation: The target dsn would result in a duplicate name in the VTOC. Processing terminates.

User response: Correct the RENAME-MASKS to prevent the creation of duplicate data set names.

**CKZ11230E RETURN CODE 8 SET FOR NOT
RENAMED DATA SET(S) | RETURN
CODE 8 SET FOR TEMPORARY DATA
SET(S)**

Explanation: A return code of 8 was requested for the indicated condition. Processing terminates.

User response: No action is required, unless a return code of 0 or 4 is desired for the condition. If a return code other than 8 is wanted, either change the CKZINI member,, or override the return code in the RENAME command.

CKZ11231I RENAMED DATA SETS:

Explanation: A RENAME SIMULATE was requested. The data set names that are listed will be renamed on the target volume.

User response: No action is required.

**CKZ11233E SYNAD TEXT: error text from
SYNADAF**

Explanation: An EXCP request failed. SYNADAF was invoked to help diagnose the problem. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ11235W ICKDSF COMMAND FAILED FOR
TARGET VOLSER: target volser

Explanation: The ICKDSF command failed. The messages from ICKDSF will be printed.

User response: If unable to determine the reason for the ICKDSF failure, contact IBM Software Support. Have available the listing that contains this message.

CKZ11243E UNABLE TO RENAME DSN=source
datasetname | USING MASK=target
mask

Explanation: The new name of a data set will exceed 44 characters. Processing terminates.

User response: Correct the RENAME-MASKS specification.

CKZ11260E ERROR DURING function FOR
VOLSER=volume - RETURN
CODE=nnnn REASON CODE=nnnn

Explanation: An error occurred using IOSCAPU, UCBLLOOK, or UCBPIN. Processing terminates.

User response: A return code 4 from UCBLLOOK may indicate the volser is offline. If unable to determine the cause of the error, contact IBM Software Support. Have available the listing containing this message.

CKZ11261E ERROR CALLING SMFWTM;
R15=nnnn

Explanation: An error occurred using the SMFWTM macro. Processing terminates.

User response: If unable to determine the cause of the error, contact IBM Software Support. Have available the listing containing this message.

CKZ11301I hh:mm:ss VVDS UPDATE STARTED
FOR VOLUME: volume - PROGRAM
REV=rrr (** SIMULATION **) |
hh:mm:ss VVDS UPDATE
COMPLETED; RETURN CODE=nnn

Explanation: Individual VVDS UPDATE processing message.

User response: No action is required.

CKZ11303I DDNAME=ddname ALLOCATED FOR
DSN=datasetname

Explanation: 'ddname' has been dynamically allocated for the indicated data set.

User response: No action is required.

CKZ11305E ALLOCATION FAILED FOR DSN:
datasetname

Explanation: Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ11305W DEALLOCATION FAILED FOR
DDNAME: ddname

Explanation: Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ11306E ERROR CALLING CKZ01VV1 tttttt
FUNCTION: function R15=nnnn
R0=nnnnnnnn LOC=lllll

Explanation: A problem occurred using a dataspace. tttttt is the name of the internal table. lllll is the location where the error occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

CKZ11307W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

CKZ11309E ERROR ACCESSING JOURNAL FILE;
LOC=lllll

Explanation: A VSAM error occurred accessing the journal file. Processing terminates.

User response: See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ11312W DATA SET MATCHES NO RENAME
MASK: datasetname (MATCHES
EXCLUDE MASK)

Explanation: The indicated data set did not match any source rename mask. If the message includes

CKZ11330E • CKZ11349E

(MATCHES EXCLUDE MASK) the data set matched an entry in the EXCLUDE-SRCNAME-MASKS keyword. Processing continues.

User response: No action is required.

CKZ11330E RETURN CODE 8 SET FOR NOT RENAMED DATA SET(S) | RETURN CODE 8 SET FOR MISSING USER CATALOG(S)

Explanation: A return code of 8 was requested for the indicated condition. Processing terminates.

User response: No action is required, unless a return code of 0 or 4 is desired for the condition. If a return code other than 8 is wanted, either change the CKZINI member, or, override the return code in the RENAME command.

CKZ11335I SMS smstypeCLAS COPIED FROM SOURCE FOR VVDS ENTRY - dsname

Explanation: No default value was given for the smstype (DATA, MGMT, or, STOR). The indicated SMS class for the entry was copied from the source data set.

User response: No action is required, unless a specific class is desired.

CKZ11340W RENAMES NOT COMPLETE FOR VVDS ENTRY - COMPONENT NAME=dsname

Explanation: A VVDS entry could not be completely renamed. Processing continues. The entry will be handled as a 'not renamed' data set.

User response: No action is required.

CKZ11341W RENAMES NOT DONE FOR VVDS ENTRY - COMPONENT NAME=dsname

Explanation: A VVDS entry could not be renamed. Processing continues. The entry will be handled as a 'not renamed' data set.

User response: No action is required.

CKZ11342W USER CATALOG NOT IN CATALOG LIST - COMPONENT NAME=dsname | BCS=user catalog

Explanation: A VVDS entry had a BCS backpointer which was not in the list of source user catalogs. Processing continues. The data set will be handled as a 'missing ucat' data set.

User response: No action is required.

CKZ11343W UNABLE TO RENAME DSN=source datasetname | USING MASK=target mask

Explanation: The new name of a data set will exceed 44 characters. Processing continues.

User response: No action is required.

CKZ11343W UNABLE TO RENAME DSN=source datasetname | USING MASK=target mask

Explanation: The new name of a data set will exceed 44 characters. Processing continues.

User response: No action is required.

CKZ11344E VVDS ENTRY NOT FOUND FOR dsname

Explanation: An expected VVDS entry was not found for the dsname. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ11345E ERROR ACCESSING VVDS, LOC=IIIII

Explanation: An error occurred processing the VVDS on the volume. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ11347E DSI CELL NOT FOUND IN VVDS RECORD FOR dsname

Explanation: An expected VVDS cell was not found for the dsname. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ11348E ERROR CALLING CKZ00902; R15=nnnn NEW DSN=target dsname

Explanation: An error occurred calling CKZ00902 to obtain ACS information for the target data set name. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ11349E VVDS WITH DSN=dataset NOT FOUND ON VOLUME=volser

Explanation: The VVDS with name of data set was not found on volume volser. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ11350E ERROR CALLING CKZ01VE2;
FUNCTION: function R15=nnnn
R0=xxxx | VVDS DSN=dataset
VOL=volser RBA=xxxx**

Explanation: An error occurred using CKZ01VE2 to read a VVDS. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ11351I RENAMED CATALOG IS NOT
USABLE: dsname**

Explanation: A catalog on a target volume was included in a RENAME-MASKS entry. The cloned catalog is not usable.

User response: No action is required.

**CKZ11352E UPDATED VVDS ENTRY EXCEEDS
MAXIMUM ALLOWABLE LENGTH**

Explanation: An updated VVDS entry was found to be larger than the maximum allowable size. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ11353I DUMP OF UPDATED VVDS ENTRY:

Explanation: A dump of the VVDS entry follows.

User response: No action is required.

**CKZ11401I hh:mm:ss VOLUME CLEANUP
STARTED - PROGRAM REV=rrr (**
SIMULATION **) | hh:mm:ss VOLUME
CLEANUP COMPLETED; RETURN
CODE=nnn**

Explanation: VOLUME CLEANUP processing message.

User response: No action is required.

**CKZ11405E ALLOCATION FAILED FOR VOLUME:
volume serial**

Explanation: Dynamic allocation for a volume failed. The associated z/OS messages are displayed. Processing terminates.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ11405W DEALLOCATION FAILED FOR
DDNAME: ddname**

Explanation: Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ11406E ERROR CALLING CKZ01VV1 tttttt
FUNCTION: function R15=nnnn
R0=nnnnnnnn LOC=lllll**

Explanation: A problem occurred using a dataspace. tttttt is the name of the internal table. lllll is the location where the error occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

**CKZ11407W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn**

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

CKZ11411E JOURNAL FILE IS EMPTY

Explanation: An attempt was made to read the journal file. The file is empty. Processing terminates.

User response: Verify that the journal file has been correctly specified.

**CKZ11430W DELETE FAILED. RETURN
CODE=nnnn REASON CODE=nnnn
VOLSER=targetvolser
DSN=sourcedatasetname**

Explanation: An attempt was made to delete a data set from a target volume. The request to SSI failed. Processing continues.

User response: The target volumes are usable. There are either unrenamed data sets or temporary data sets that were not deleted.

CKZ11435I PROCESSING VOLSER=targetvolser

Explanation: Volume cleanup informational message.

User response: No action is required.

**CKZ11436I DELETE WILL BE ISSUED FOR
DSN=sourcedatasetname**

Explanation: During a RENAME SIMULATE run, informational message issued for data sets which will be deleted from the target volumes during the actual RENAME run.

User response: No action is required.

**CKZ11501I hh:mm:ss VOLUME RESTORES
STARTED - PROGRAM REV=rrr |
hh:mm:ss VOLUME RESTORES
COMPLETED; RETURN CODE=nnn**

Explanation: VOLUME RESTORES processing message.

User response: No action is required.

**CKZ11503I DDNAME=ddname ALLOCATED FOR
DSN=datasetname**

Explanation: 'ddname' has been dynamically allocated for the indicated data set.

User response: No action is required.

CKZ11504E OPEN FAILED FOR DDNAME=ddname

Explanation: 'ddname' was allocated for DB2 Cloning Tool to use, but, the open for the file failed. Processing terminates.

User response: If unable to determine the reason the open failed, contact IBM Software Support. Have available the listing that contains this message.

**CKZ11505E ALLOCATION FAILED FOR DSN:
datasetname**

Explanation: Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ11505W DEALLOCATION FAILED FOR
DDNAME: ddname**

Explanation: Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ11506E ERROR CALLING CKZ01VV1 tttttt
FUNCTION: function R15=nnnn
R0=nnnnnnnn LOC=lllll**

Explanation: A problem occurred using a dataspace. tttttt is the name of the internal table. lllll is the location where the error occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

**CKZ11507W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn**

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

**CKZ11523E ERROR OPENING DDNAME
DRSTATS**

Explanation: An error occurred opening sysout file DRSTATS. Processing terminates.

User response: Check that the appropriate DD statement is in the JCL. If unable to determine the cause of the error, contact IBM Software Support. Have available the listing that contains this message.

**CKZ11524E VOLUME RESTORE FAILED FOR
volser. SEE SYSOUT DRSTATS FOR
MORE INFORMATION**

Explanation: An error occurred during RERUN mode attempting to restore a volume. Processing terminates.

User response: Check the DRSTATS for the error that occurred. If unable to correct the problem, contact IBM Software Support.

**CKZ11531W REVERTING TO USE VOLBKUP
DIRECTLY**

Explanation: An error occurred using a subset of the VOLBKUP file. The VOLBKUP file will be used rather than a subset for volume metadata restores. Processing continues.

User response: Contact IBM Software Support. Have available the listing containing this message.

**CKZ11532E VOLUMES IN VOLBKUP NOT IN
EXPECTED SEQUENCE**

Explanation: An error occurred using the VOLBKUP file. The volume data in the VOLBKUP file was not in the expected sequence. Processing terminates.

User response: Contact IBM Software Support. Have

available the listing containing this message.

CKZ11533E UNEXPECTED END OF FILE ON VOLBKUP

Explanation: An error occurred using the VOLBKUP file. Processing terminates.

User response: Contact IBM Software Support. Have available the listing containing this message.

CKZ11560E ERROR DURING function FOR VOLSER=volume - RETURN CODE=nnnn REASON CODE=nnnn

Explanation: An error occurred using IOSCAPU, UCBLLOOK, or UCBPIN. Processing terminates.

User response: A return code 4 from UCBLLOOK may indicate the volser is offline. If unable to determine the cause of the error, contact IBM Software Support. Have available the listing containing this message.

CKZ11601I hh:mm:ss CHECK TARGETS STARTED - PROGRAM REV=rrr | hh:mm:ss CHECK TARGETS COMPLETED; RETURN CODE=nnn

Explanation: CHECK TARGETS processing message.

User response: No action is required.

CKZ11607W ERROR CALLING CKZ01HEX; FUNCTION: function R15=nnnn

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

CKZ11609E ERROR ACCESSING JOURNAL FILE; LOC=IIIII

Explanation: A VSAM error occurred accessing the journal file. Processing terminates.

User response: See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ11611E JOURNAL VOLUME PAIR RECORD(S) NOT FOUND

Explanation: An expected record was not found in the DB2 Cloning Tool journal file. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ11660E UCBCAN ERROR; RETURN CODE=nn REASON CODE=nn | UCINFO ERROR; RETURN CODE=nn REASON CODE=nn

Explanation: An error occurred using UCBCAN or UCINFO. Processing terminates.

User response: Contact IBM Software Support. Have available the listing containing this message.

CKZ11676E DEVICE TYPE AND MODEL NOT FOUND FOR volser

Explanation: IOSCDs for the volser did not return information about the target volume.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ11679E TARGET VOLSER volser WAS NOT FOUND ONLINE

Explanation: The indicated target volume serial was not found online on the system running RENAME. Processing terminates.

User response: Check that all the target volumes are online to the system which will be running RENAME.

CKZ11681E VOLUME volser DEVICE NUMBER nnnn HAS NO chpid PATHS AVAILABLE

Explanation: The indicated device has no paths available for DB2 Cloning Tool to use to access the volume. Processing terminates.

User response: Ensure that at least one chpid is online for each target device specified.

CKZ11701I hh:mm:ss CATALOG VCLOSE STARTED - PROGRAM REV=rrr (SIMULATION **) | hh:mm:ss CATALOG VCLOSE COMPLETED; RETURN CODE=nnn**

Explanation: CATALOG VCLOSE processing message.

User response: No action is required.

CKZ11703I DDNAME=ddname ALLOCATED FOR DSN=datasetname

Explanation: 'ddname' has been dynamically allocated for the indicated data set.

User response: No action is required.

CKZ11704E OPEN FAILED FOR DDNAME=ddname

Explanation: 'ddname' was allocated for DB2 Cloning Tool to use, but, the open for the file failed. Processing terminates.

User response: If unable to determine the reason the open failed, contact IBM Software Support. Have available the listing that contains this message.

CKZ11705W ALLOCATION FAILED FOR DDNAME: ddname | DEALLOCATION FAILED FOR DDNAME: ddname

Explanation: Dynamic allocation for a data set failed, or, dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. If an allocation failure occurs, processing continues without using CATALOG VCLOSE services. If a deallocation failure occurs, processing continues.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ11707W ERROR CALLING CKZ01HEX; FUNCTION= function R15=nnnn

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

CKZ11720I CONSOLE name ACQUIRED FOR CATALOG VCLOSE COMMANDS

Explanation: A console session has been acquired so that RENAME can issue MODIFY CATALOG,VCLOSE commands.

User response: No action is required.

CKZ11721I CONSOLE name FREED

Explanation: The operator console has been freed.

User response: No action is required.

CKZ11722W UNABLE TO ACQUIRE A CONSOLE

Explanation: RENAME has failed to acquire a console for performing operator commands. Processing continues without using a console to perform operator commands.

User response: No action is required.

CKZ11723I CONSOLE name ALREADY IN USE, WILL TRY ANOTHER

Explanation: The displayed name is already in use, probably from another copy of RENAME. RENAME will increment the number portion of the name and try again.

User response: No action is required.

CKZ11724W MCS ALERT RECEIVED; text

Explanation: An alert has been received for the console. Text describes the type of alert. Processing continues.

User response: No action is required.

CKZ11727I NON RESPONSE MDBS RECEIVED: nnn

Explanation: Informational message that displays the number of received messages that were not a response to the command issued.

User response: No action is required.

CKZ11731W MCSOPER ERROR; FUNCTION= function RC=nnnn RSN=nnnn | MCSOPMSG ERROR; FUNCTION= function RC=nnnn RSN=nnnn

Explanation: An error occurred using MCSOPER or MCSOPMSG. Processing continues.

User response: Contact IBM Software Support. Have available the listing containing this message.

CKZ11740I COMMAND: text

Explanation: Display operator command being issued.

User response: No action is required.

CKZ11741W WAIT TIME EXCEEDED FOR COMMAND RESPONSE

Explanation: A response to the operator command was not received in a timely manner. Processing continues.

User response: Determine if the catalog address space (CAS) is not responding to modify commands or is unable to process them in a timely manner. If unable to determine the cause, please report this message to IBM Software Support. Have available the listing containing this message.

**CKZ12500I PGM CKZ00125 invoked to perform
???????? function on VOL=volser
UNIT=addr - PROGRAM REV=rrr**

Explanation: Program CKZ00125 is acknowledging a request to DUMP or RESTORE the VTOC and VVDS of the indicated volume.

User response: No action is required.

**CKZ12501E CKZ00125 Parameter Error. parameter
description**

Explanation: CKZK00125 has detected invalid parameters from the calling routine. This is an internal error.

User response: Contact IBM Software Support. Have the execution output listing available.

**CKZ12504E DSPSERV CREATE Error
RC=xx,RSN=yy, requesting nnnnn 4K
Datspace Blocks.**

Explanation: Dataspace Creation failed with the above Return and Reason codes.

User response: Contact IBM Software Support. Have the execution output listing available.

**CKZ12505E ALESERV ADD Error R15=xx Creating
Datspace Alet**

Explanation: An error occurred while attempting to add an entry into the DU-AL for a private dataspace that has been created.

User response: Contact IBM Software Support. Have the execution output listing available.

**CKZ12506I PGM CKZ00125 ???????? Processing
Completed RC=xx timestamp**

Explanation: Program CKZ00125 processing is terminating with the above return-code.

User response: If RC=00, None. If the Return-Code is any non-zero value, then contact IBM Software Support, and have the execution output listing available. There will be previous messages indicating the error causing the bad return code.

**CKZ12507E ERROR CALLING CKZ01VV1 tttttt
FUNCTION: function R15=nnnn
R0=nnnnnnnn LOC=lllll**

Explanation: A problem occurred using a dataspace. tttttt is the name of the internal table. lllll is the location where the error occurred. Processing terminates.

User response: Contact IBM Software Support. Have

available the listing that contains this message and the CKZINI member.

**CKZ12508W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn**

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

**CKZ12510E CKZ00125 ABENDED S-xxx | CKZ00125
ABENDED U-xxxx**

Explanation: Program CKZ00125 has suffered an abend and is taking appropriate recovery and cleanup actions. The requested function appearing in the CKZ12500I message has failed.

User response: Contact IBM Software Support. Have the execution listings and the SYSUDUMP output available. Also, make note of the MVS operating system release, and the type of hardware that was being accessed.

**CKZ12520E I/O Error Reading Volume Label on
Device /xxxx**

Explanation: Program CKZK00125 was unable to read the volume label at the indicated device address.

User response: Determine if the device at the indicated address can be varied OFFLINE and ONLINE. The volume may be uninitialized. If the volume can be successfully mounted, then contact IBM Software Support. Have the execution output listing available. Also, make note of the MVS operating system release, and the type of hardware that was being accessed.

**CKZ12521E Validation on Device /xxxx failed.
Detected VOL=yyyyyy**

Explanation: Program CKZ00125 read the volume label at the indicated device address and found a volume serial number that was different than what was expected.

User response: Contact IBM Software Support. Have the execution output listing available.

**CKZ12531E SYSVTOC Shr Reserve Required. |
SYSZVVDS Shr Reserve Required. | **
Backup is Fuzzy ****

Explanation: While preparing to DUMP the VTOC and/or VVDS of the volume indicated by the CKZ12500I message, adequate RESERVE resources were not held. This is an internal error.

User response: Contact IBM Software Support. Have the execution output listing available.

CKZ12533E DataSpace size is Insufficient.

Explanation: While preparing to DUMP the VTOC and/or VVDS of the volume indicated by the CKZ12500I message, program CKZ00125 was unable to allocate a private dataspace of a sufficient size.

User response: Contact IBM Software Support. Have the execution output listing available. Also, make note of the MVS operating system release.

CKZ12535I nnnnnnn Records Written Backup File

Explanation: Informational.

User response: No action is required.

**CKZ12540E SYSVTOC Excl Reserve Required. |
SYSZVVDS Excl Reserve Required. | **
Volume Restore Not Attempted ****

Explanation: While preparing to RESTORE the VTOC and/or VVDS of the volume indicated by the CKZ12500I message, adequate RESERVE resources were not held. This is an internal error.

User response: Contact IBM Software Support. Have the execution output listing available.

CKZ12541I volser Volume Restore SUCCESSFUL.

Explanation: Informational.

User response: No action is required.

**CKZ12542E ** volser Volume Restore FAILED ** |
** WARNING: Volume volser may be
UNUSABLE ****

Explanation: RESTORE the VTOC and/or VVDS of the indicated volume was unsuccessful. If integrity of the volume is at risk, the second warning message is also issued.

User response: Contact IBM Software Support, and have the execution output listing available. There will be previous messages indicating the error causing the failure detection. Also, make note of the MVS operating system release, and the type of hardware that was being accessed.

**CKZ12544E nnnnnnn Records Examined | Backup
Data for VOL=yyyyyy not found**

Explanation: While preparing to RESTORE the VTOC and/or VVDS of the indicated volume, CKZ00125 could not locate the dump records in the sequential backup dataset.

User response: Contact IBM Software Support. Have the execution output listing available. If possible, save the backup dataset indicated by this message in case it is requested by IBM Software Support.

**CKZ12545E Record ID nnnnnnn Out of Sequence. |
Incomplete Backup Data for
VOL=vvvvvv | EOF Encountered after
Record ID nnnnnnn | Invalid Backup
Version: nnn Expected: nnn**

Explanation: While preparing to RESTORE the VTOC and/or VVDS of the indicated volume, CKZ00125 has determined that the logical contents of the sequential backup data set are invalid. "Invalid Backup Version" can happen if the version or maintenance level has changed for module CKZ00125 between the prior RENAME SAFE and this RENAME RERUN.

User response: Contact IBM Software Support. Have the execution output listing available. If possible, save the backup data set indicated by this message in case it is requested by IBM Software Support.

**CKZ12546I VOL=vvvvvv Dataspace Load Complete:
timestamp**

Explanation: Informational.

User response: No action is required.

**CKZ12547E Error Allocating ????? bytes for record
????? of ???? | Dataspace size ?????-K is
insufficient. | ?????-K used up to this
point.**

Explanation: While preparing to RESTORE the VTOC and/or VVDS of the indicated volume, CKZ00125 exceeded a predetermined dataspace size.

User response: Contact IBM Software Support. Have the execution output listing available. If possible, save the backup data set indicated by this message in case it is requested by IBM Software Support.

CKZ12548E Invalid Dump Record. ID ????????

Explanation: While preparing to RESTORE the VTOC and/or VVDS of the indicated volume, CKZ00125 has determined that the logical contents of the sequential backup data set are invalid.

User response: Contact IBM Software Support. Have the execution output listing available. If possible, save the backup data set indicated by this message in case it is requested by IBM Software Support.

CKZ12549E Buffer Capacity Exceeded. TYPE=????

Explanation: RESTORE processing has failed due to incorrect buffer size calculations. This is an internal error.

User response: Contact IBM Software Support, and have the execution output listing available. Also, make note of the MVS operating system release, and the type of hardware that was being accessed.

**CKZ12550E ** EXCP I/O ERROR processing the
????????? ** | Track(CCHH) Address:
cchh | Synad: SYNAD error text**

Explanation: An uncorrectable error has occurred to the device being DUMPed or RESTORed while CKZ00125 was performing I/O using the EXCP access method.

User response: Contact IBM Software Support, and have the execution output listing available. Also, make note of the MVS operating system release, and the type of hardware that was being accessed.

CKZ12554E Storage Subsystem for Device ???? Does not Support ECKD CCWs.

Explanation: CKZ00125 has detected an old technology DASD Control Unit that does not support hardware features that are minimally required by the DB2 Cloning Tool product. Such control units are typically used for supporting devices that pre-date 3380's. All control units for 3390's support ECKD transfer protocol.

User response: No action is required. DB2 Cloning Tool cannot be used for this device.

CKZ12555E Invalid track format

Explanation: CKZ00125 has detected a track on the DASD volume that does not have the expected format.

User response: Verify that the DASD volume is in a valid copy relationship. This error might be caused by a volume copied outside of DB2 Cloning Tool where the volume was not actually copied and the last copy of the volume was done with the FCNOCOPY option.

CKZ12560I *variable text*

Explanation: Informational Statistics.

User response: No action is required.

CKZ12561I *variable text*

Explanation: Informational statistics regarding DUMP processing.

User response: No action is required.

CKZ12561W Note: Requested VVDS Dataset not in use.

Explanation: Informational warning regarding DUMP processing. The caller of CKZ00125 specified a non-standard data set name for the VVDS, which was not found on the volume. The correct data set name for the VVDS was found, and will be assumed as valid for DUMP processing.

User response: No action is required. DUMP processing continues.

CKZ12561E ** ERROR ** Requested VVDS Dataset Not Found. | ** ERROR ** Required VVDS Dataset Not Found.

Explanation: A VVDS was not found on the volume that was being processed for dump. In addition, the volume was SMS managed, and/or contained VSAM data sets.

User response: Verify that the volume is usable. If not, then a volume restore is in order. In either case, contact IBM Software Support, and have the execution output listings available.

CKZ12562I *variable text*

Explanation: Informational statistics regarding RESTORE processing.

User response: No action is required.

CKZ12562E Volume Dump for ?????? is Unusable.

Explanation: While attempting to RESTORE the VTOC and VVDS, program CKZ00125 detected that the location of the VTOC does not match the CCHHR address at the time of the DUMP.

User response: If the DUMP backup data set is current, then contact IBM Software Support, and have the execution output listing available.

CKZ12562E Volume Dump for ?????? is Unusable.

Explanation: While attempting to RESTORE the VTOC and VVDS, program CKZ00125 detected that the location of the VTOC does not match the CCHHR address at the time of the DUMP.

User response: If the DUMP backup data set is current, then contact IBM Software Support, and have the execution output listing available.

CKZ12578I *variable text*

Explanation: Informational diagnostics.

User response: No action is required.

CKZ12599I *variable text*

Explanation: Informational messages typically used for performance feedback purposes.

User response: No action is required.

CKZ13001I hh:mm:ss SORTS STARTED -
PROGRAM REV=rrr | hh:mm:ss
SORTS COMPLETED; RETURN
CODE=nnn

Explanation: BCS backup sorts processing message.

User response: No action is required.

CKZ13003I DDNAME=ddname ALLOCATED FOR
DSN=datasetname

Explanation: 'ddname' has been dynamically allocated for the indicated data set.

User response: No action is required.

CKZ13004E DDNAME MISSING: ddname | OPEN
FAILED FOR DDNAME=ddname

Explanation: 'ddname' was allocated for DB2 Cloning Tool to use, but, the open for the file failed. Processing terminates.

User response: If unable to determine the reason the open failed, contact IBM Software Support. Have available the listing that contains this message.

CKZ13005E ALLOCATION FAILED FOR DSN:
datasetname

Explanation: Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ13005W DEALLOCATION FAILED FOR
DDNAME: ddname

Explanation: Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ13006E ERROR CALLING CKZ01VV1 tttttt
FUNCTION: function R15=nnnn
R0=nnnnnnnn LOC=llll

Explanation: A problem occurred using a daspace. tttttt is the name of the internal table. llll is the location where the error occurred. Processing terminates.

User response: Contact IBM Software Support. Have

available the listing that contains this message and the CKZINI member.

CKZ13007W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

CKZ13009E ERROR ACCESSING JOURNAL FILE;
LOC=lllll

Explanation: A VSAM error occurred accessing the journal file. Processing terminates.

User response: See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ13011E JOURNAL USER CATALOG
RECORD(S) NOT FOUND

Explanation: An expected record was not found in the DB2 Cloning Tool journal file. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ13012E UNABLE TO ESTABLISH ESTAEX;
R15=nnnn LOC=lllll

Explanation: A subroutine was not able to establish an estaex environment. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ13030I SORT FOR SOURCE BCS RECORDS
SUCCESSFUL; BCS=bcs dsname

Explanation: The sort of the indicated BCS's records was successful.

User response: No action is required.

CKZ13031E SORT FOR SOURCE BCS RECORDS
FAILED; R15=nnnn; BCS=bcs dsname

Explanation: The sort of the indicated BCS's records failed. Processing terminates.

User response: Correct the problem that caused SORT to fail. If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ13041W NO BCS ENTRIES MATCH RENAME CRITERIA FOR SOURCE BCS=BCS dsname

Explanation: No BCS record keys matched the source dsname(s)/mask(s) specified. Processing continues.

User response: Verify that the correct USERCATALOGS were specified in the COPY command and that the correct source names were specified in the RENAME-MASKS.

CKZ13042W DUPLICATE KEYS IN BACKUP; FIRST RECORD KEPT

Explanation: A record with a duplicate key was detected from the backup file. The records either do not contain a creation date or the creation dates are equal. The second record was dropped. Both records are printed. Processing continues.

User response: Examine the printed records. The dropped record may have the desired data.

CKZ13042W DUPLICATE KEYS IN BACKUP; FIRST RECORD KEPT

Explanation: A record with a duplicate key was detected from the backup file. The records either do not contain a creation date or the creation dates are equal. The second record was dropped. Both records are printed. Processing continues.

User response: Examine the printed records. The dropped record may have the desired data.

CKZ13042W DUPLICATE KEYS IN BACKUP; FIRST RECORD KEPT

Explanation: A record with a duplicate key was detected from the backup file. The records either do not contain a creation date or the creation dates are equal. The second record was dropped. Both records are printed. Processing continues.

User response: Examine the printed records. The dropped record may have the desired data.

CKZ13043E BCS BACKUPS DID NOT COMPLETE; CHECK COPY STEP

Explanation: The journal indicates that the BCS backups did not complete. Processing terminates.

User response: Check that the COPY command has completed successfully before initiating the RENAME command.

CKZ13044E UNEXPECTED EOF ON BCS BACKUP FILE; BCS=bcs dsname

Explanation: A logical 'end of data' condition is expected; a physical end of file was detected.

User response: Check that the COPY command has completed successfully before initiating the RENAME command.

CKZ13045E NO BCS ENTRIES MATCH RENAME CRITERIA FOR ANY SOURCE BCS

Explanation: No BCS record keys matched the source dsname(s)/mask(s) specified. Processing terminates.

User response: Verify that the correct USERCATALOGS were specified in the COPY command and that the correct source names were specified in the RENAME-MASKS.

CKZ13099E ABEND DURING BCS SORT

Explanation: An abend occurred for a BCS SORT. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ13501I hh:mm:ss BCS CLEANUP STARTED - PROGRAM REV=rrr (SIMULATION **) | hh:mm:ss BCS CLEANUP COMPLETED; RETURN CODE=nnn**

Explanation: BCS CLEANUP processing message.

User response: No action is required.

CKZ13503I DDNAME=ddname ALLOCATED FOR DSN=datasetname

Explanation: 'ddname' has been dynamically allocated for the indicated data set.

User response: No action is required.

CKZ13504E OPEN FAILED FOR DDNAME=BCSRECS

Explanation: BCSRECS did not open successfully. Processing terminates.

User response: Check that //BCSRECS points to the correct data set. If unable to determine the reason for the failure, contact IBM Software Support. Have available the listing containing this message.

CKZ13505E ALLOCATION FAILED FOR DSN: datasetname

Explanation: Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

CKZ13505W • CKZ13560I

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ13505W DEALLOCATION FAILED FOR DDNAME: ddname

Explanation: Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ13507W ERROR CALLING CKZ01HEX; FUNCTION: function R15=nnnn

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

CKZ13509E ERROR ACCESSING JOURNAL FILE; LOC=IIIII

Explanation: A VSAM error occurred accessing the journal file. Processing terminates.

User response: See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ13511E JOURNAL CONTROL RECORD NOT FOUND

Explanation: An expected record was not found in the DB2 Cloning Tool journal file. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ13520W KEY OF ZEROS FOUND; ENTRY BYPASSED

Explanation: An entry was found in BCSRECS with a key of zeros. The entry is bypassed. Processing continues.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ13521E KEY OF ZEROS RETURNED

Explanation: The BCS entry with a key of zeros (binary) was returned; it was not requested. Processing terminates.

User response: Contact IBM Software Support. Have

available the listing that contains this message.

CKZ13522I RECORDS DELETED FROM BCS(S), COUNT=nnn

Explanation: The indicated number of BCS records have been removed from the target BCS(s).

User response: No action is required.

CKZ13523I ENTRIES DELETED BY SVC26, COUNT=nnn

Explanation: The indicated number of BCS entries have been removed from the target BCS(s) using SVC26.

User response: No action is required.

CKZ13545E ERROR ACCESSING BCS=bcs dsname; LOC=IIIII

Explanation: A VSAM error occurred accessing the indicated BCS. Processing terminates.

User response: See associated CKZERRnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ13549E DELETE FAILED; R15=nnn REASON CODE=rrr MODULE=mm DSN=dsname

Explanation: The SVC26 delete for a data set failed.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ13599E ABEND DURING DB2 CLONING TOOL PROCESSING

Explanation: An abend occurred during BCS cleanup. Processing terminates.

User response: Determine the reason for the abend and correct if possible. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ13560I WAITING FOR EXCLUSIVE CONTROL OF BCS bcs name

Explanation: The BCS is currently in use by another job. The wait will continue until the BCS is no longer in use by another job or the wait time limit is exceeded. Processing continues.

User response: No action is required.

CKZ13561E UNABLE TO ALLOCATE BCS: bcs name; WAIT TIME LIMIT EXCEEDED

Explanation: The wait for exclusive control of the BCS has exceeded the wait time limit. Processing terminates.

User response: Change the scheduling of the jobs so the DB2 Cloning Tool job does not run when another job has the BCS allocated. Or increase the wait time limit so the DB2 Cloning Tool job can wait longer for the other job to terminate. The wait time limit is set by the CKZINI parameter CONCURRENT_EXECUTIONS_WAIT_TIME.

CKZ13601I Catalog orphan cleanup started - program rev=REV ** SIMULATION ** | Catalog orphan cleanup started - return code= return_code.

Explanation: This message indicates one of the following:

- In SIMULATION mode, it indicates the start of the BCSCLEAN catalog orphan phase for keywords CLEANUP-CATALOG-ORPHANS and CLEANUP-CATALOG-DSNMASKS. The string ** SIMULATION ** is appended to the message text.
- In a normal run, this message indicates the end of the BCSCLEAN catalog orphan phase with given return code.

User response: No action is required.

CKZ13603I DD name= ddname allocated for user catalog user_catalog.

Explanation: The user catalog listed in the message text has been dynamically allocated to the DD name listed in the message.

User response: No action is required.

CKZ13604I DD name= ddname allocated for volume volume.

Explanation: The volume specified in the messages text has been dynamically allocated to the DD name *ddname*.

User response: No action is required.

CKZ13605E Allocation failed for user catalog user_catalog.

Explanation: Dynamic allocation of the user catalog listed in the message failed. Processing stops.

User response: Verify that the user catalog name is correct and accessible. Check for other error messages that may have been issued. If unable to determine the cause of this error, contact IBM Software Support.

CKZ13605W Deallocation failed for DD name: ddname.

Explanation: Dynamic deallocation of the DD name listed in the message text failed.

User response: Check for other error messages that may have been issued. If unable to determine the cause of this error, contact IBM Software Support.

CKZ13608E Allocation failed for volume volume.

Explanation: Dynamic allocation of the volume listed in the message text failed. Processing stops.

User response: Verify that the volume name is correct and accessible. Check for other error messages that may have been issued. If unable to determine the cause of this error, contact IBM Software Support.

CKZ13609E Error accessing journal file; LOC= internal_location.

Explanation: An error occurred accessing the journal file. Processing stops.

User response: Contact IBM Software Support. Have available the journal file and the listing that contains these messages.

CKZ13610I DD name= ddname deallocated.

Explanation: The DD name listed in the message text has been deallocated.

User response: No action is required.

CKZ13611E Journal record_type record error_details.

Explanation: The record type listed in the message produced an error that is described by the string *error_details*. Processing stops.

User response: Contact IBM Software Support. Have available the journal file and the listing that contains these messages.

CKZ13612E Journal volume pair record count does not match count found.

Explanation: The number of volume pair records found does not match the expected number from the control record. Processing stops.

User response: Contact IBM Software Support. Have available the journal file and the listing that contains these messages.

CKZ13615I **Target user catalogs for this BCSCLEAN job:** *list*.

Explanation: All user catalogs from the journal are listed and will be included in this execution.

User response: No action is required.

CKZ13616I **Target volumes for this BCSCLEAN job:** *list*.

Explanation: All target volumes from the journal are listed and will be included in this execution.

User response: No action is required.

CKZ13617E **SVC99 error encountered for dsn:** *data_set_name* – **details follow:**

Explanation: Dynamic allocation failed for the given data set name. A detailed set of dynamic allocation error messages (up to nine detail lines) follows this message. Processing stops.

User response: Evaluate the error messages and try to resolve the error condition. If unable to resolve the issue, contact IBM Software Support. Have available the listing that contains these messages.

CKZ13618I **Skipping type with dsn *data_set_name*: extension records not supported.**

Explanation: The catalog record for the given type and data set name spans across multiple physical catalog records, which is not supported. Processing continues, but this data set is ignored.

User response: Contact IBM Software Support. Have available the listing that contains these messages.

CKZ13619I **No orphan records were found - no deletes are required.**

Explanation: No orphan data sets were found that require action.

User response: No action is required.

CKZ13620I **Total orphan records checked:** *number_of_records*.

Explanation: This message lists the total number of catalog records that were examined for volume or data set name mask comparison or both.

User response: No action is required.

CKZ13621I **Orphan records checked in user catalog:** *number_of_records*.

Explanation: This message lists the number of catalog records within the current user catalog that were

examined for volume or data set name mask comparison or both.

User response: No action is required.

CKZ13622I **Total orphan records deleted:** *number_of_records*.

Explanation: This message lists the total number of catalog records that were deleted due to a match on target volume per the CLEANUP-CATALOG-ORPHANS keyword.

User response: No action is required.

CKZ13623I **Total orphan dsnmask entries deleted:** *number_of_records*.

Explanation: This message lists the total number of files deleted that matched any CATALOG-CLEANUP-DSNMASK data set name mask value.

User response: No action is required.

CKZ13624I **Start of user catalog processing:** *user_catalog_name*.

Explanation: The user catalog listed in the message is open for processing.

User response: No action is required.

CKZ13625I **End of user catalog processing:** *user_catalog_name*.

Explanation: The user catalog listed in the message is closed and processing is complete.

User response: No action is required.

CKZ13628I **Start of delete processing .**

Explanation: This message indicates the start of the delete phase. The files identified by CLEANUP-CATALOG-ORPHANS (if specified) will be deleted with the NOSCRATCH option. The files identified by CLEANUP-CATALOG-DSNMASK will be deleted with the SCRATCH option if they reside on a non-target volume; they will be deleted with the NOSCRATCH option if they reside on a target volume.

User response: No action is required.

CKZ13629I **End of delete processing.**

Explanation: This message indicates the end of the delete phase.

User response: No action is required.

| **CKZ13631E** AMSOPEN failed. R15=*return_code*.

| **Explanation:** The AMSOPEN macro failed with the given return code. Processing stops.

| **User response:** Contact IBM Software Support. Have available the journal file and the listing that contains these messages.

| **CKZ13632I** Non-zero returned by IDCAMS;
RC=*return_code*.

| **Explanation:** IDCAMS returned an unexpected non-zero return code, as shown in message. Processing stops.

| **User response:** Contact IBM Software Support. Have available the journal file and the listing that contains these messages.

| **CKZ13645E** Error accessing BCS=*user_catalog* loc=
internal_location.

| **Explanation:** The user catalog (BCS) could not be read at the internal location listed in the message text.

| **User response:** Contact IBM Software Support. Have available the journal file and the listing that contains these messages.

| **CKZ13648I** Delete: *record_type data_set_name sim*
diagnostic_information.

| **Explanation:** The file *data_set_name* with the catalog record type *record_type* was deleted. If '** SIMULATION **' appears in the message text, no delete is performed. Additional diagnostic information may be included in the message text (*diagnostic_information*).

| **User response:** No action is required.

| **CKZ13660I** Waiting for exclusive control of BCS
user_catalog.

| **Explanation:** Contention was encountered when trying to allocate the user catalog. Processing waits for the interval specified in the value of INI keyword CONCURRENT_EXECUTIONS_WAIT_TIME.

| **User response:** No action is required.

| **CKZ13661E** Unable to allocate BCS *user_catalog*; wait
time limit exceeded.

| **Explanation:** Allocation failed for the user catalog due to contention and a subsequent exhaustion of a retry loop, as directed by the current value of INI keyword CONCURRENT_EXECUTIONS_WAIT_TIME.

| **User response:** Determine the source of the contention and the process that is holding the resource. Consider lengthening the time interval for the INI keyword CONCURRENT_EXECUTIONS_WAIT_TIME. If no

| resolution can be found, contact IBM Software Support. Have available the journal file and the listing that contains these messages.

CKZ14001I hh:mm:ss BCS UPDATES STARTED -
PROGRAM REV=rrr (** SIMULATION
**) | hh:mm:ss BCS UPDATES
COMPLETED; RETURN CODE=nnn

Explanation: BCS update processing message.

User response: No action is required.

CKZ14006E ERROR CALLING CKZ01VV1 tttttt
FUNCTION: function R15=nnnn
R0=nnnnnnnn LOC=lllll

Explanation: A problem occurred using a dataspace. tttttt is the name of the internal table. lllll is the location where the error occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

CKZ14007W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

CKZ14008E UNABLE TO LOAD PROGRAM:
program name | UNABLE TO LINK TO
PROGRAM: program name

Explanation: The indicated program name was not found. Processing terminates.

User response: Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

CKZ14009E ERROR ACCESSING JOURNAL FILE;
LOC=lllll

Explanation: A VSAM error occurred accessing the journal file. Processing terminates.

User response: See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ14011E JOURNAL CONTROL RECORD NOT
FOUND

Explanation: An expected record was not found in the DB2 Cloning Tool journal file. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ14030I BCS UPDATE COMPLETED; RETURN CODE=nn SYSOUT DD=sysout ddname FOR TARGET BCS=bcs dsname | BCS UPDATE FAILED ; RETURN CODE=nn SYSOUT DD=sysout ddname FOR TARGET BCS=bcs dsname

Explanation: The BCS update for the indicated target BCS has ended.

User response: No action is required if the RETURN CODE is zero. If the RETURN CODE is not zero, check the indicated sysout file for warning or error messages related to the BCS update.

CKZ14031I DB2RECS UPDATED FOR TARGET BCS=

Explanation: The DB2 Catalog and DB2 Directory data sets entries have been written to the DB2RECS file.

User response: No action is required.

CKZ14032E NO DATA SETS MATCHED THE SPECIFIED DB2PREFIX: db2prefix

Explanation: No data sets for any target usercatalog matched the high-level qualifier specified in DB2PREFIX. No entries were written to DB2RECS. Processing terminates.

User response: Correct the name specified in the DB2PREFIX keyword.

CKZ14033W BCS UPDATE BYPASSED FOR TARGET BCS=bcs dsname ; error text SOURCE BCS=bcs dsname

Explanation: The BCS update for the indicated target/source BCS has not been done. The error text indicates the reason.

User response: No action is required.

CKZ14040E UNEXPECTED CONDITION; error text

Explanation: An unexpected condition occurred while dispatching a BCS update task or while waiting for the completion of a BCS update task. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ14041E SORT FOR BCS RECORDS DID NOT COMPLETE SUCCESSFULLY; BCS=bcs dsname

Explanation: The sort of the records for the indicated BCS was not successful. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ14089I NOT ALL DATASETS HAVE BEEN CATALOGED

Explanation: An error occurred during BCS rename processing that caused some data sets to not be cataloged in a target catalog. Message CKZ14030I will indicate which catalog had an error.

User response: Correct the cause of the error and rerun RENAME if possible.

CKZ14101I hh:mm:ss BCS UPDATE STARTED - PROGRAM REV=rrr (** SIMULATION **) | hh:mm:ss BCS UPDATE COMPLETED; RETURN CODE=n RECORD COUNT=nnn

Explanation: BCS UPDATE task processing message.

User response: No action is required.

CKZ14103I DDNAME=ddname ALLOCATED FOR DSN=datasetname

Explanation: 'ddname' has been dynamically allocated for the indicated data set.

User response: No action is required.

CKZ14104E OPEN FAILED FOR DDNAME=ddname

Explanation: 'ddname' was allocated for DB2 Cloning Tool to use, but, the open for the file failed. Processing terminates.

User response: If unable to determine the reason the open failed, contact IBM Software Support. Have available the listing that contains this message.

CKZ14105E ALLOCATION FAILED FOR DSN: datasetname | ALLOCATION FAILED FOR DDNAME: ddname

Explanation: Dynamic allocation for a data set or ddname failed. The associated z/OS messages are displayed. Processing terminates.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ14105W DEALLOCATION FAILED FOR DDNAME: ddname

Explanation: Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response: If unable to determine the reason for

the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ14106E ERROR CALLING CKZ01VV1 tttttt
FUNCTION: function R15=nnnn
R0=nnnnnnnn LOC=llll**

Explanation: A problem occurred using a dataspace. tttttt is the name of the internal table. llll is the location where the error occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

**CKZ14107W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn**

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

**CKZ14109E ERROR ACCESSING JOURNAL FILE;
LOC=llll**

Explanation: A VSAM error occurred accessing the journal file. Processing terminates.

User response: See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ14110E DUPLICATE JOURNAL ENTRY;
LOC=llll**

Explanation: A duplicate record was detected. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ14111E JOURNAL USER CATALOG
RECORD(S) NOT FOUND**

Explanation: An expected record was not found in the DB2 Cloning Tool journal file. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ14112E UNABLE TO ESTABLISH ESTAEX;
R15=nnnn**

Explanation: The program was not able to establish an estaex environment. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ14122I IDCAMS WILL BE USED FOR THIS
BCS**

Explanation: Informational message. BCS extension records will cause program CKZ00150 to invoke IDCAMS.

User response: No action is required.

**CKZ14123I EXPIRATION DATE IGNORED FOR
GDG ENTRY; GDG BASE NAME=gdg
base name**

Explanation: Informational message. The expiration date of the GDG base has been ignored.

User response: No action is required.

**CKZ14135I SMS smstypeCLAS COPIED FROM
SOURCE FOR BCS ENTRY - dsname**

Explanation: No default value was given for the smstype (DATA, MGMT, or, STOR). The indicated SMS class for the entry was copied from the source data set.

User response: No action is required, unless a specific class is desired.

CKZ14140E UNEXPECTED CONDITION; error text

Explanation: An unexpected condition occurred. 'error text' has a description of the problem. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ14141I BCS ENTRY NOT ON SOURCE
VOLUME SERIALS; BCS KEY=bcs key
name**

Explanation: A BCS entry matched the rename masks, but, the entry is not cataloged on the source volume serials.

User response: No action is required.

**CKZ14142E BCS ENTRY PARTIALLY ON SOURCE
VOLUME SERIALS; BCS KEY DSN=bcs
key name | GDG HAS GDS ENTRIES
THAT ARE NOT ON SOURCE
VOLUME SERIALS; GDG BASE
NAME=gdg base name**

Explanation: The indicated BCS entry is only partially on the source volume serials. Processing terminates.

User response: Ensure that data sets (VSAM spheres, GDSs associated with a base GDG) are wholly contained on the source volume serials.

**CKZ14142W GDG HAS MIGRATED GDS ENTRIES;
GDG BASE NAME=*gdg base name* |
GDG HAS TAPE GDS ENTRIES; GDG
BASE NAME=*gdg base name***

Explanation: The indicated BCS entry is only partially on the source volume serials. Processing continues, but the target GDSs which are migrated or on tape will not be accessible.

User response: Ensure that data sets (VSAM spheres, GDSs associated with a base GDG) are wholly contained on the source volume serials.

CKZ14143E UNABLE TO RENAME DSN=*source datasetname* USING MASK=*target rename mask*

Explanation: The new name of a data set will exceed 44 characters, or, the new name of a GDG base will exceed 35 characters. Processing terminates.

User response: Correct the RENAME-MASKS specification.

**CKZ14144E NO SOURCE DSN/MASK MATCH
FOUND FOR DSN=*datasetname* ENTRY
IS PART OF BCS KEY=*bcs key name***

Explanation: A BCS entry did not fully match the specified rename masks. Processing terminates.

User response: Ensure the RENAME-MASKS specification includes all spheres of VSAM components and all aliases for non-VSAM entries.

**CKZ14145E ERROR ACCESSING BCS=*bcs dsname*;
LOC=*lcccc***

Explanation: A VSAM error occurred accessing the indicated BCS. Processing terminates.

User response: See associated CKZERRnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ14146W BCS ENTRY SKIPPED; RECORD TYPE
NOT SUPPORTED**

Explanation: Probably, a 'U' type BCS entry was detected. The entry is printed. DB2 Cloning Tool does not support renaming user catalogs on the target volume serials.

User response: No action is required.

**CKZ14147E DUPLICATE BCS ENTRY | AN ENTRY
FOR *name* ALREADY EXISTS IN THE
TARGET BCS**

Explanation: A duplicate record was detected when adding an entry to the target user catalog. The entry

already exists in the target user catalog. The existing entry could be there from a prior run of DB2 Cloning Tool where BCSCLEAN was not used to remove the entry, or the RENAME-MASKS being used caused duplicate data set names to be created, or the existing entry could belong to a data set that was created prior to the DB2 Cloning Tool run and is not on a target volume. The duplicate catalog entry is printed. Processing terminates.

User response: Ensure the RENAME-MASKS are not causing duplicate data set names to be created. If the RENAME-MASKS are not causing duplicate data set names, determine why the entry already exists in the target user catalog. To replace existing entries in the target user catalog, use the RECATALOG(Y) keyword in the RENAME command.

**CKZ14148E ERROR CALLING CKZ00902;
R15=*nnnn* NEW DSN=*new datasetname***

Explanation: An error occurred invoking the ACS routines for the indicated new data set name. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ14149E FAILED TO POSITION IN BCSRECS

Explanation: An error occurred attempting to position to the correct entry to update in the BCSRECS file. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ14150W GDG IS EMPTY; GDG BASE
NAME=*gdg base name* | ALL GDG'S
ARE MIGRATED; GDG BASE
NAME=*gdg base name***

Explanation: An empty GDG was encountered and GDG-EMPTY(RETAIN,RC(4)) was specified or all GDGs are migrated and GDG-ALL-MIGRATED(RETAIN,RC(4)) was specified.

User response: No action is required.

**CKZ14151E UNEXPECTED END OF FILE
ENCOUNTERED ON BCSRECS**

Explanation: An error occurred attempting to read the entry to update in the BCSRECS file. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ14160I WAITING FOR EXCLUSIVE CONTROL OF BCS bcs name

Explanation: The BCS is currently in use by another job. The wait will continue until the BCS is no longer in use by another job or the wait time limit is exceeded. Processing continues.

User response: No action is required.

CKZ14161E UNABLE TO ALLOCATE BCS: bcs name; WAIT TIME LIMIT EXCEEDED

Explanation: The wait for exclusive control of the BCS has exceeded the wait time limit. Processing terminates.

User response: Change the scheduling of the jobs so the DB2 Cloning Tool job does not run when another job has the BCS allocated. Or increase the wait time limit so the DB2 Cloning Tool job can wait longer for the other job to terminate. The wait time limit is set by the CKZINI parameter CONCURRENT_EXECUTIONS_WAIT_TIME.

CKZ14199E ABEND DURING BCS UPDATE

Explanation: An abend occurred for a BCS update task. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ14201I hh:mm:ss BCSRECS FORMAT STARTED - PROGRAM REV=rrr | hh:mm:ss BCSRECS FORMAT COMPLETED; RETURN CODE=nnn

Explanation: BCSRECS file processing message.

User response: No action is required.

CKZ14204E OPEN FAILED FOR DDNAME=BCSRECS

Explanation: BCSRECS did not open successfully. Processing terminates.

User response: Check that //BCSRECS points to the correct data set. If unable to determine the reason for the failure, contact IBM Software Support. Have available the listing that contains this message.

CKZ14240E ERROR WRITING TO BCSRECS; R15=nnnn

Explanation: An error occurred formatting the BCSRECS file. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ15001I hh:mm:ss IDCAMS PROCESS STARTED - PROGRAM REV=rrr (SIMULATION **) | hh:mm:ss IDCAMS PROCESS COMPLETED; RETURN CODE=nnn ENTRIES RECATALOGED=nnn**

Explanation: BCS IDCAMS processing message.

User response: No action is required.

CKZ15003I DDNAME=ddname ALLOCATED FOR DSN=datasetname

Explanation: 'ddname' has been dynamically allocated for the indicated data set.

User response: No action is required.

CKZ15005E ALLOCATION FAILED FOR DSN: datasetname

Explanation: Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ15005W DEALLOCATION FAILED FOR DDNAME: ddname

Explanation: Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ15007W ERROR CALLING CKZ01HEX; FUNCTION: function R15=nnnn

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

CKZ15009E ERROR ACCESSING JOURNAL FILE; LOC=lllll

Explanation: A VSAM error occurred accessing the journal file. Processing terminates.

User response: See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ15011E JOURNAL CONTROL RECORD NOT FOUND | JOURNAL IDC RECORD(S) NOT FOUND

Explanation: An expected record was not found in the DB2 Cloning Tool journal file. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ15030E RETURN CODE 8 SET FOR ORPHANCATENTRY

Explanation: A return code of 8 was requested for the indicated condition. Processing terminates.

User response: No action is required, unless a return code of 0 or 4 is desired for the condition. If a return code other than 8 is wanted, either change the CKZINI member, or, override the return code in the RENAME command.

CKZ15031E AMSOPEN FAILED; R15=nnnn

Explanation: An attempt was made to issue an IDCAMS command. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ15032I NON-ZERO RETURNED BY IDCAMS; RC=nnnn

Explanation: An IDCAMS command failed with return code nnnn. The IDCAMS messages are displayed. Processing will continue if ORPHANCATENTRY return code is less than or equal to 4. Processing will terminate if ORPHAN-CATENTRY return code is greater than 4.

User response: No action is required.

CKZ15035E DEVICE TYPE NOT RECOGNIZED; DEVICE=X'ddddddd'; SOURCE DSN=source datasetname

Explanation: The device type could not be converted to one for use by IDCAMS RECATALOG. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ15040E UNEXPECTED CONDITION; error text

Explanation: An unexpected condition occurred. 'error text' has a description of the problem. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ15041E MORE THAN nnn IDCAMS MESSAGES

Explanation: More than nnn messages were returned for an IDCAMS command. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ15045E ERROR ACCESSING VVDS=vvds dsname; LOC=IIIII

Explanation: A VSAM error occurred accessing the indicated VVDS. Processing terminates.

User response: See associated CKZERRnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ15046W ENTRY NOT FOUND IN VVDS; DSN=dsname

Explanation: A VSAM component was not found in the VVDS. The entry cannot be recataloged. Processing will continue if ORPHANCATENTRY return code is less than or equal to 4. Processing will terminate if ORPHANCATENTRY return code is greater than 4.

User response: No action is required.

CKZ15047W ENTRY COULD NOT BE RECATALOGED; DSN=dsname

Explanation: A data set was not found on the target volume serials and cannot be recataloged. Processing will continue if ORPHANCATENTRY return code is less than or equal to 4. Processing will terminate if ORPHANCATENTRY return code is greater than 4.

User response: No action is required.

CKZ15048W ENTRY SKIPPED DUE TO CLUSTER FAILURE; TYPE=type DSN=dsname

Explanation: A type AIX® or PATH entry could not be recataloged because the associated base cluster could not be recataloged.

User response: No action is required.

CKZ15049E DELETE FAILED; R15=nnn REASON CODE=rrr MODULE=mm DSN=dsname

Explanation: The SVC26 delete for a data set failed.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ15101I hh:mm:ss UPDATE IAM ASSOCIATIONS PROCESS STARTED - PROGRAM REV=rrr (** SIMULATION **) | hh:mm:ss UPDATE IAM ASSOCIATIONS PROCESS COMPLETED; RETURN CODE=nnn ENTRIES UPDATED=nnn

Explanation: Update IAM associations processing message.

User response: No action is required.

CKZ15103I DDNAME=ddname ALLOCATED FOR DSN=datasetname

Explanation: ddname has been dynamically allocated for the indicated data set.

User response: No action is required.

CKZ15104E macro FAILED FOR DDNAME=ddname loc=lllll

Explanation: The OPEN or RDJFCB z/OS macro failed for the identified ddname. Processing terminates.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ15105E ALLOCATION FAILED FOR DSN: datasetname

Explanation: Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ15105W DEALLOCATION FAILED FOR DDNAME: ddname

Explanation: Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ15106E ERROR CALLING CKZ01VV2 tttttt FUNCTION: function R15=nnnn R0=nnnnnnnn LOC=lllll

Explanation: A problem occurred using a dataspace. tttttt is the name of the internal table. lllll is the

location where the error occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing containing these messages and the CKZINI member.

CKZ15107W ERROR CALLING CKZ01HEX; FUNCTION: function R15=nnnn

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

CKZ15109E ERROR ACCESSING JOURNAL FILE; LOC=lllll

Explanation: A VSAM error occurred accessing the journal file. Processing terminates.

User response: See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ15111E JOURNAL UIAM RECORD(S) NOT FOUND | JOURNAL UIAM RECORD IS WRONG VERSION

Explanation: An expected record was not found in the DB2 Cloning Tool journal file. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains these messages.

CKZ151131E AMSOPEN FAILED; R15=nnnn

Explanation: An attempt was made to issue an IDCAMS command. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains these messages.

CKZ151132I NON-ZERO RETURNED BY IDCAMS; RC=nnnn

Explanation: An IDCAMS command failed with return code nnnn. The IDCAMS messages are displayed. Processing will continue if the return code is 4 or less. Processing will terminate if the return code is 8 or above.

User response: For return code 4 or less, none. For return code 8 or above see displayed IDC messages. If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ151134E UNEXPECTED DATA FOUND IN IAMPRINT reason

Explanation: The parsing of the data in the IAMPRINT dd failed. The data in the IAMPRINT dd is different than expected. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains these messages.

CKZ151141E MORE THAN nnn IDCAMS MESSAGES

Explanation: More than nnn messages were returned for an IDCAMS command. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains these messages.

CKZ151145E ASSOCIATED type FOR PATH DSN=datasetname NOT PRESENT type SOURCE DSN=datasetname

Explanation: The AIX or CLUSTER data set associated with the identified PATH does not exist. It is possible to get this error when using RERUN when the rename mask entries that cover IAM data sets have changed. Processing terminates.

User response: Verify the AIX or CLUSTER data set is on a cloned volume and is being renamed. If RERUN is being used and the rename mask entries that cover IAM data sets have changed, it may be necessary to start the cloning over from the beginning. If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ15146E ASSOCIATED CLUSTER FOR AIX DSN=datasetname NOT PRESENT CLUSTER SOURCE DSN=datasetname

Explanation: The CLUSTER data set associated with the identified AIX does not exist. It is possible to get this error when using RERUN when the rename mask entries that cover IAM data sets have changed. Processing terminates.

User response: Verify the CLUSTER data set is on a cloned volume and is being renamed. If RERUN is being used and the rename mask entries that cover IAM data sets have changed, it may be necessary to start the cloning over from the beginning. If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ20001I Stored procedure completed; Return code=return_code

Explanation: The CKZ stored procedure has completed processing. *return_code* is the return code from the stored procedure.

User response: No action is required.

CKZ20003I DDNAME: ddname allocated for DSN: data_set_name

Explanation: *ddname* has been dynamically allocated for the indicated data set.

User response: No action is required.

CKZ20004E Open failed for DDNAME: ddname DSN: data_set_name reason

Explanation: The open of *ddname* has failed. Reason will be given if there is additional information about the failure. Processing terminates.

User response: Verify the data set has the correct attributes.

CKZ20005E Allocation failed for DSN: data_set_name

Explanation: Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ20005W Deallocation failed for DDNAME: ddname

Explanation: Dynamic deallocation for a *ddname* failed. The associated z/OS messages are displayed. Processing continues.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ20007W Error calling CKZ01HEX; Function: function R15=value

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

CKZ20008E macro failed for program: program_name RC=X'return_code' RSN=X'reason_code'

Explanation: The LINK or LOAD of the indicated program has failed. Processing terminates.

User response: Check that the //STEPLIB of the address space in which the stored procedure is running is correct.

CKZ20010E Invalid type specified: *type*

Explanation: The stored procedure has been invoked with an unknown TYPE= request. Processing terminates.

User response: Verify that a correct TYPE= request is being made to the stored procedure.

CKZ20011E macro error; RC: return_code RSN:
reason_code

Explanation: A call of the indicated system macro has failed. *return_code* is the return code and *reason_code* is the reason code from the macro call. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ20016E Invalid parameter; parameter Record number: *record_number DSN: data_set_name reason*

Explanation: The identified parameter is invalid. The parameter is located on the indicated record number in the indicated parameter data set. *reason* indicates why the parameter is invalid. Processing terminates.

User response: Correct the parameter.

CKZ20017E Invalid continuation; Record number:
record_number DSN: data_set_name

Explanation: The identified record number in the indicated data set has invalid continuation. Processing terminates.

User response: Correct the record.

CKZ20018E Too many continuations; Max =
maximum Record number: record_number
DSN: *data_set_name*

Explanation: The number of continuation records exceeds the maximum allowed. Processing terminates.

User response: Reduce the number of continuation records.

CKZ20019E LRECL not 80 for DSN: *data_set_name*

Explanation: The identified data set does not have an LRECL of 80. Processing terminates.

User response: Change the data set to have an LRECL of 80 or use a different data set that has an LRECL of 80.

CKZ20020E Error calling routine FC=*feedback_code*
LOC=*location*

Explanation: An error has been returned from a call to a Language Environment® service. *routine* identifies the service returning the error. *feedback_code* is the feedback code from the service in hexadecimal. *location* is the location where the error occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ20103I DDNAME: *ddname allocated for DSN: data_set_name*

Explanation: *ddname* has been dynamically allocated for the indicated data set.

User response: No action is required.

CKZ20104E Open failed for DDNAME: *ddname*
DSN: *data_set_name reason*

Explanation: The open of *ddname* has failed. A reason will be supplied if there is additional information about the failure. Processing terminates.

User response: Verify that the data set has the correct attributes.

CKZ20105E Allocation failed for DSN: *data_set_name*

Explanation: Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ20105W Deallocation failed for DDNAME:
ddname

Explanation: Dynamic deallocation for a *ddname* failed. The associated z/OS messages are displayed. Processing continues.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ20107W Error calling CKZ01HEX; Function:
function R15=value

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

CKZ20108E **Unable to load program:** *program_name*
RC=X'return_code' **RSN=X'reason_code'**

Explanation: The LOAD of the indicated program has failed. Processing terminates.

User response: Check that the //STEPLIB of the address space in which the stored procedure is running is correct.

CKZ20109E **Internal error; LOC:** *location reason*

Explanation: An internal error has occurred. *location* is the location where the error occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ20110E **Invalid type specified:** *type*

Explanation: The stored procedure has been invoked with an unknown TYPE= request. Processing terminates.

User response: Verify a correct TYPE= request is being made to the stored procedure.

CKZ20111E **macro error; RC:** *return_code* **RSN:**
reason_code

Explanation: A call of the indicated system macro has failed. *return_code* is the return code and *reason_code* is the reason code from the macro call. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ20119E **LRECL not 80 for DSN:** *data_set_name*

Explanation: The identified data set does not have an LRECL of 80. Processing terminates.

User response: Change the data set to have an LRECL of 80 or use a different data set that has an LRECL of 80.

CKZ20120E **Invalid type parameter; Record number:**
record_number **DSN:** *data_set_name*
Keyword: *keyword reason*

Explanation: The identified parameter of the indicated type is invalid. The parameter is located on the indicated record number in the parameter data set listed in *data_set_name*. *reason* indicates why the parameter is invalid. Additional information may be given to help qualify the context of the parameter. Processing terminates.

User response: Correct the parameter.

CKZ20121E **Missing type parameter; DSN:**
data_set_name **Keyword:** *keyword*

Explanation: The identified parameter of the indicated type is required. The parameter is missing from the indicated parameter data set. Additional information may be given to help qualify the context of the missing parameter. Processing terminates.

User response: Add the missing parameter.

CKZ20122E **SSID: *ssid* is not defined in the system parameters data set; DSN:** *data_set_name*

Explanation: The identified *ssid* was referenced in the cloning, but the subsystem is not defined in the system parameters data set. *Data_set_name* identifies the system parameters data set being used. Processing terminates.

User response: Verify the correct system parameters data set is being used. Verify the SSID is defined in the system parameters data set.

CKZ20123E *parameter1* **cannot be used with**
parameter2

Explanation: *parameter1* is mutually exclusive with *parameter2*. Only one of the parameters can be specified. Processing terminates.

User response: Remove one of the parameters from the parameter file.

CKZ20124I **CLONING-TYPE = OFFLINE was specified for SLB source; Will use CLONING-TYPE = ONLINE instead**

Explanation: A cloning type of offline was specified for a cloning from an SLB. An SLB cloning is an online cloning, so cloning type of online will be used instead. Processing terminates.

User response: Specify CLONING-TYPE = ONLINE when cloning from a SLB.

CKZ20130E **Call to DSNTIAR failed; RC:** *return_code*

Explanation: A call to DSNTIAR failed. *return_code* is the return code from DSNTIAR. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ20131E **SQL call failed; SQLCODE:** *sql_code*
SQLSTATE: *sql_state*

Explanation: An error has been returned from a SQL call. Processing terminates.

User response: If unable to determine the reason for the failure, contact IBM Software Support. Have available the listing that contains this message.

CKZ20132E Call to administrative task scheduler stored procedure failed; RC: *return_code*
Loc: *location* **SP Name:** *stored_procedure_name* **Task name:** *task_name*

Explanation: An error has been returned from call to an ADMIN_TASK stored procedure. Any messages returned by the ADMIN_TASK stored procedure will be displayed. Processing terminates.

User response: If unable to determine the reason for the failure, contact IBM Software Support. Have available the listing that contains this message.

CKZ20133E Unable to add task to administrative task scheduler; Task name is already in use **Task name:** *task_name*

Explanation: The addition of a task to the administrative task scheduler failed because the task name of the task being added is already in use. The task name may belong to another cloning or a REMOVE was not done to remove the tasks before a BUILD was done. Processing terminates.

User response: Verify the task name is not in use by another cloning. If the task name should not be use, it will be necessary to manually delete the tasks from the administrative task scheduler.

CKZ20134E Unable to update task in administrative task scheduler; Task does not exist **Loc:** *location* **Task name:** *task_name*

Explanation: The update of a task in the administrative task scheduler failed because the task name of the task being updated did not exist. This error can happen if a CLONE or RECLONE is done after a BUILDJCL. CLONE and RECLONE can only be done after a BUILD. BUILDJCL only builds JCL and does not add any tasks to the administrative task scheduler. Or, the task may have been deleted manually and not by a REMOVE. Processing terminates.

User response: If a BUILDJCL and not a BUILD was done, do a REMOVE, a BUILD, and then the CLONE or RECLONE. If the task was removed manually do a REMOVE, a BUILD, and then the CLONE or RECLONE.

CKZ20140E Status file is empty; DSN: *data_set_name*

Explanation: An attempt to read the status file found it to be empty. Processing terminates.

User response: Verify the correct status file data set is being used, the correct request type has been used, and the status file has not been altered. If unable to determine the reason for the failure, contact IBM Software Support. Have available the listing that contains this message.

CKZ20141E Delete of JCL member failed; *macro* RC: *return_code* DSN: *data_set_name* **Member:** *member_name*

Explanation: An attempt to delete a JCL member failed. *macro* indicates the system macro that gave the error and *return_code* is the return code from the macro. Processing terminates.

User response: If unable to determine the reason for the failure, contact IBM Software Support. Have available the listing that contains this message.

CKZ20142E DB2_PLAN not found in CKZINI; DSN: *data_set_name*

Explanation: The DB2_PLAN keyword was not found in CKZINI. BIND-ON-TARGET=Y has been specified and the plan name in CKZINI is needed to perform the plan bind. Processing terminates.

User response: Verify the CKZINI parameter in the product parameter file points to a valid CKZINI.

CKZ20143E DB2_PLAN in CKZINI is invalid; *reason* DSN: *data_set_name*

Explanation: The DB2_PLAN keyword found in CKZINI is invalid. *reason* indicates why it is invalid. Processing terminates.

User response: Verify the CKZINI parameter in the product parameter file points to a valid CKZINI.

CKZ20150E Wait time exceeded for Task Name: *task_name* **Loc:** *location* *reason*

Explanation: A cloning job did not complete within the expected time. *reason* indicates the wait reason. Processing terminates.

User response: Verify the administrative task scheduler is functioning and the z/OS system is not delaying the execution of the cloning jobs. If unable to determine the reason for the failure, contact IBM Software Support. Have available the listing that contains this message.

CKZ20151E Job completed with an error; Task Name: *task_name* **Max rc:** *return_code* **Job id:** *job_id* **Completion type:** *type*

Explanation: A cloning job completed with an abend or an unexpected return code. The cloning stops. *job_id* indicates the JES job ID of the cloning job. The abend code will be displayed if the cloning job abended. Processing terminates.

User response: Use the job ID to look at the SYSOUT of the failed cloning job to identify the problem. If the JCL needs to be regenerated, the cloning will need to be cleaned up (CLEAN) and removed (REMOVE). Then update the parameter files, BUILD, and then CLONE. If

| the JCL does not need to be regenerated, the cloning
| will need to be cleaned up (CLEAN) and then rerun
| (CLONE or RECLONE).

| **CKZ20152E Unexpected job status: Task Name:**
| *task_name* **Loc: location Unable to**
| **continue monitoring cloning jobs**

| **Explanation:** The administrative task scheduler
| returned an unexpected status for the indicated task.
| The administrative task scheduler may have been
| unable to run the task or the administrative task
| scheduler may have been shut down while the task
| was running. Processing terminates.

| **User response:** Verify the administrative task
| scheduler is functioning. If unable to determine the
| reason for the failure, contact IBM Software Support.
| Have available the listing that contains this message.

CKZ21501I hh:mm:ss DB2GETBACKINFO
STARTED - PROGRAM REV=rrr |
hh:mm:ss DB2GETBACKINFO
COMPLETED; RETURN CODE=nnn

Explanation: DB2GETBACKINFO processing message.

User response: No action is required.

CKZ21504E DDNAME MISSING: ddname | OPEN
FAILED FOR DDNAME: ddname

Explanation: 'ddname' was specified for DB2 Cloning
Tool to use. Processing terminates.

User response: Either correct the ddname specified, or
add the appropriate ddname to the job's JCL.

CKZ21505E ALLOCATION FAILED FOR DSN:
datasetname | DEALLOCATION
FAILED FOR DSN: datasetname

Explanation: Dynamic allocation for a data set failed,
or, dynamic deallocation for a ddname failed. The
associated z/OS messages are displayed. Processing
terminates.

User response: If unable to determine the reason for
the failure from the associated z/OS messages, contact
IBM Software Support. Have available the listing
containing these messages.

CKZ21507W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn

Explanation: An error occurred using CKZ01HEX to
print a record. Processing continues.

User response: Please report this message to IBM
Software Support.

CKZ21508E UNABLE TO LOAD PROGRAM:
program name

Explanation: The indicated program name was not
found. Processing terminates.

User response: Check that the job's //STEPLIB library
is correct. If unable to resolve the problem, contact IBM
Software Support.

CKZ21509E ERROR ACCESSING BSDS FILE;
LOC=IIIIII

Explanation: A VSAM error occurred accessing the
BSDS file. Processing terminates.

User response: See associated CKZVSEnnE error
messages. If unable to resolve problem, contact IBM
Software Support. Have available the listing that
contains these messages.

CKZ21523I BACKUP DUMPTAPES FOR
LOCATION: location

| **Explanation:** A list of SLB dump tapes follows.

| **User response:** No action is required.

CKZ21530I nnnn type RECORDS WRITTEN

Explanation: Displays the number of type records
written to the backinfo data set.

User response: No action is required.

CKZ21531E NO BACKUPS FOUND FOR
LOCATION: location | NO BACKUPS
FOUND FOR LOCATION: location AND
DUMPCLASS dump_class |
REQUESTED BACKUP NOT USABLE -
NO DUMPTAPES FOUND

Explanation: No DB2 BACKUP SYSTEM backups
were found in HSM for the indicated location. If the
USE-DUMPTAPES keyword was supplied with
DUMP-CLASS, and that dump class was not found, the
requested dump class is listed in the message. For
USE-DUMPTAPES, this message might be generated
when the backup has an HSM DUMPSTATE that is not
ALLCOMPLETE, or at least one dump tape for the
backup has an expiration date prior to the current date;
those backups are excluded from consideration.

User response: Verify the correct location is being
used . For USE-DUMPTAPES, ensure that the desired
backup has dump tapes associated with it, that none of
the tapes have expired, and that the dumpstate is
ALLCOMPLETE. For USE-DUMPTAPES and
DUMP-CLASS, ensure that the DUMP-CLASS keyword
is correctly specified and that a usable backup exists for
that DUMP-CLASS.

**CKZ21532E NO BACKUPS FOUND FOR TOKEN:
token**

Explanation: No DB2 BACKUP SYSTEM backups were found in HSM for the indicated token.

User response: Verify the correct token is being used.

**CKZ21533I BACKUPS FOR LOCATION: location
list of backups**

Explanation: Displays the DB2 BACKUP SYSTEM backups that were found in HSM for the indicated location.

User response: No action is required.

**CKZ21534W UNEXPECTED DATA FOUND IN HSM
RESPONSE**

Explanation: The data found in the HSM response was not as expected.

User response: Contact IBM Software Support. Have available the listing that contains these messages.

CKZ21535I SELECTED BACKUP

Explanation: Displays information about the selected backup.

User response: No action is required.

**CKZ21536E Some backup volumes are not on
DASD; VOLSER: *****.**

Explanation: Some of the backup volumes for the selected backup have a volser of *****. A volser of ***** indicates that the backup volume is not on DASD. The selected backup cannot be used if any backup volume is not on DASD. Processing terminates.

User response: Select a backup that has all the backup volumes on DASD.

**CKZ21537E BSDS RECORD NOT FOUND FOR
record-type**

Explanation: A record of the indicated type was not found in the BSDS.

User response: Verify the BSDS being used is valid. The LOCATION keyword can be used instead of BSDS-DDN. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages and a copy of the BSDS being used.

**CKZ21538I BSDS LOCATION: location WILL BE
USED**

Explanation: Informational message that displays the location obtained from the BSDS.

User response: No action is required.

**CKZ21539E LOCATE ERROR; RETURN
CODE=nnnn X'hhhh' REASON
CODE=nnnn X'hhhh'**

Explanation: A LOCATE request failed. The return code and reason code are displayed in decimal and hexadecimal format. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ21540E CKZ00900 UNEXPECTED RESULTS;
error text**

Explanation: An unexpected condition occurred calling program CKZ00900. 'error text' has a description of the problem. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ21541E ARCHSEND ERROR; RETURN
CODE=nnnn X'hhhh' REASON
CODE=nnnn X'hhhh'**

Explanation: An ARCHSEND request failed. The return code and reason code are displayed in decimal and hexadecimal format. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ21543E MULTIPLE DUMP CLASSES FOUND
FOR SELECTED VERSION: *dump_class1*
AND *dump_class2* – DUMP-CLASS
KEYWORD REQUIRED**

Explanation: The DUMP-CLASS keyword was not specified, but because multiple dump classes were found, the DUMP-CLASS keyword is required.

User response: Provide the DUMP-CLASS keyword, or select a different backup version that only has one dump class by using the TOKEN or LAST keywords.

**CKZ21545I USERCATALOG: user-catalog WAS
FOUND ON BACKUP VOLUME:
vvvvvv**

Explanation: The indicated user catalog was found on the indicated backup volume.

User response: No action is required.

CKZ21546E USERCATALOG: user-catalog WAS NOT FOUND ON ANY BACKUP VOLUME

Explanation: The indicated user catalog was not found on any backup volume. The indicated user catalog may have been created after the backup that created the backup volumes. Processing terminates.

User response: Verify the correct user catalogs have been specified in the USERCATALOGS keyword.

CKZ21547E UNABLE TO DETERMINE VOLSER FOR USERCATALOG: user-catalog

Explanation: It is not possible to determine the current volsr of the indicated user catalog. The user catalog may not exist. Processing terminates.

User response: Verify the user catalog exists. If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ21548E USERCATALOG: user-catalog IS NOT ON A COPIED VOLUME: vvvvvv

Explanation: The indicated user catalog was found to be currently on a volume that is not one of the source volumes. Processing terminates.

User response: Verify the correct user catalogs have been specified in the USERCATALOGS keyword.

CKZ21549I USERCATALOG: user-catalog WAS NOT FOUND ON EXPECTED BACKUP VOLUME: vvvvvv | USERCATALOG MAY HAVE MOVED SINCE BACKUP THAT CREATED BACKUP VOLUMES | SCANNING BACKUP VOLUMES TO FIND USERCATALOG

Explanation: The indicated user catalog was not found on the expected backup volume. The user catalog may have moved since the backup was run that created the backup volumes. The backup volumes will be scanned to find the user catalog. This process may take several minutes depending on the number of backup volumes that need to be scanned.

User response: No action is required.

CKZ21551E REQUIRED KEYWORD MISSING: keyword

Explanation: A keyword required for processing has been omitted. Processing terminates.

User response: Specify the required keyword.

CKZ21552E THE SAME DDNAME HAS BEEN SPECIFIED FOR MULTIPLE KEYWORDS: ddname

Explanation: The indicated ddname has been specified in multiple keywords. The specified ddnames must all be different. Processing terminates.

User response: Specify different ddnames in the keywords.

CKZ21553E KEYWORD: keyword MAXIMUM LENGTH: nnn EXCEEDED

Explanation: The operand entered for a keyword exceeded the maximum length allowed for the operand. nnn is the maximum allowed length for the keyword. Processing terminates.

User response: Correct the length of the keyword's operand.

CKZ21554E KEYWORD HAS MORE THAN 1 OPERAND; ONLY 1 ALLOWED: keyword

Explanation: Multiple operands were detected for a keyword; only one operand is permitted. Processing terminates.

User response: Correct the keyword to use one operand.

CKZ21555E DUMP-CLASS KEYWORD REQUIRES USE-DUMPTAPES KEYWORD

Explanation: The DUMP-CLASS keyword was specified, but the USE-DUMPTAPES keywords was not specified. USE-DUMPTAPES is required with the DUMP-CLASS keyword.

User response: Provide the USE-DUMPTAPES keyword and resubmit.

CKZ21556E NOTHING SPECIFIED FOR KEYWORD: keyword

Explanation: A keyword was entered without an appropriate operand. Processing terminates.

User response: Specify an appropriate operand for the keyword.

CKZ21557E DUPLICATE FOUND; KEYWORD: keyword ENTRY: entry

Explanation: The indicated entry for the keyword was previously specified. Processing terminates.

User response: Remove the duplicate entry.

CKZ21558E INVALID VALUE IN KEYWORD:
keyword VALUE: value error text

Explanation: The value in the keyword is invalid. 'error text' indicates the problem detected with the value. Processing terminates.

User response: Correct the value specified in the keyword.

CKZ21560E UCBLOOK ERROR; RETURN
CODE=nn REASON CODE=nn
LOC=lllll

Explanation: An error occurred during UCBLOOK processing. Processing terminates.

User response: Contact IBM Software Support. Have available the listing containing this message.

CKZ21569E WORK-DSN: dataset-name DOES NOT
EXIST OR IS NOT AVAILABLE FOR
USE

Explanation: The dynamic allocation of the indicated data set failed. The data set may not exist or may be currently allocated to another job. Processing terminates.

User response: Verify the data set exists and is not currently allocated to another job.

CKZ21581I DSNS FOR KEYWORD: keyword list of
dsns

Explanation: Parsing found the listed dsns for the keyword.

User response: No action is required.

CKZ21586I VALIDATING KEYWORD: keyword

Explanation: Parsing is checking the indicated keyword in the command.

User response: No action is required.

CKZ22001I hh:mm:ss DB2UPDATE STARTED -
PROGRAM REV=rrr | hh:mm:ss
DB2UPDATE COMPLETED; RETURN
CODE=nnn

Explanation: DB2UPDATE command processing message.

User response: No action is required.

CKZ22003I DDNAME=ddname ALLOCATED FOR
DSN=datasetname

Explanation: 'ddname' has been dynamically allocated for the indicated data set.

User response: No action is required.

CKZ22004E DDNAME missing: ddname | Open
failed for DDNAME: ddname

Explanation: ddname was specified for DB2 Cloning Tool to use. Processing terminates.

User response: Either correct the DDNAME specified, or, add the appropriate DDNAME to the job's JCL.

CKZ22005E ALLOCATION FAILED FOR DSN:
datasetname

Explanation: Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ22005W DEALLOCATION FAILED FOR
DDNAME: ddname

Explanation: Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ22006E ERROR CALLING CKZ01VV1 tttttt
FUNCTION: function R15=nnnn
R0=nnnnnnnn LOC=lllll

Explanation: A problem occurred using a dataspace. tttttt is the name of the internal table. lllll is the location where the error occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

CKZ22007W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

CKZ22008E UNABLE TO LOAD PROGRAM:
program name

Explanation: The indicated program name was not found. Processing terminates.

User response: Check that the job's //STEPLIB library

is correct. If unable to resolve the problem, contact IBM Software Support.

**CKZ22009E ERROR ACCESSING JOURNAL FILE;
LOC=IIIII**

Explanation: A VSAM error occurred accessing the journal file. Processing terminates.

User response: See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ22010E DUPLICATE JOURNAL ENTRY;
LOC=IIIII**

Explanation: A duplicate record was detected. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ22011E JOURNAL CONTROL RECORD NOT
FOUND | JOURNAL DSN MASK
RECORD(S) NOT FOUND | JOURNAL
VOLUME PAIR RECORD(S) NOT
FOUND | JOURNAL DB2 CONTROL
RECORD NOT FOUND | JOURNAL
DB2 HLQ RECORD(S) NOT FOUND |
JOURNAL DB2 STOGROUP
RECORD(s) NOT FOUND**

Explanation: An expected record was not found in the DB2 Cloning Tool journal file. Processing terminates.

User response: Verify the same value was used for the DB2-NAME keyword that was used with the prior DB2UPDATE command for this DB2 subsystem or data sharing group. If unable to resolve the problem, Contact IBM Software Support. Have available the listing that contains this message.

**CKZ22012E JOURNAL CONTROL RECORD IS
WRONG VERSION | JOURNAL DB2
CONTROL RECORD IS WRONG
VERSION | JOURNAL VOLP RECORD
IS WRONG VERSION | JOURNAL
XMSK RECORD IS WRONG VERSION**

Explanation: The journal record does not match the expected format. The record is printed. Processing terminates.

User response: Verify that different releases of DB2 Cloning Tool have not been run using the same journal data set. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains this message.

**CKZ22013E RECORD COUNT IS ZERO; LOC=IIIII
| COUNT MISMATCH; RECORDS
READ=rrrr CONTROL RECORD
COUNT=cccc LOC=IIIII**

Explanation: There was a problem with the journal records needed to initiate the DB2 update. For the first format, the journal control record indicate no entries were added. For the second format, the number of records read from the journal, rrrr, is not the same as the number indicated in the journal control record, cccc. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ22015E THE COPY PROCESS DID NOT
COMPLETE SUCCESSFULLY**

Explanation: The journal indicates that the COPY command did not complete successfully. Processing terminates.

User response: No action is required.

**CKZ22016E DB2-HLQS ENTRIES MISMATCH;
PRIMARY MEMBER HAD DIFFERENT
DB2-HLQS ENTRIES**

Explanation: The entries entered for this member in keyword DB2-HLQS does not match the entries specified for the primary member's run of DB2UPDATE. The entries for the primary member's run are printed. Processing terminates.

User response: Correct the entries to match those specified for the primary member.

**CKZ22017E STOGROUPS ENTRIES MISMATCH;
PRIMARY MEMBER HAD DIFFERENT
STOGROUPS ENTRIES**

Explanation: The entries entered for this member in keyword STOGROUPS does not match the entries specified for the primary member's run of DB2UPDATE. The entries for the primary member's run are printed. Processing terminates.

User response: Correct the entries to match those specified for the primary member.

**CKZ22030I OPTIONS IN EFFECT FOR THIS
EXECUTION: merged options**

Explanation: Informational message indicating how DB2UPDATE will handle the options. The displayed options are derived from the CKZINI and any overriding specifications in the command input.

User response: No action is required.

CKZ22031E PRIOR COPY WAS A SIMULATION | PRIOR RENAME WAS A SIMULATION | PRIOR RENAME WAS NOT RUN

Explanation: The journal indicates that the COPY command or the RENAME command was a simulation, or the RENAME command has not been run. Processing terminates.

User response: Run DB2UPDATE after both the COPY and RENAME have successfully run in non-simulation.

CKZ22032E INTERNAL ERROR; FIELD TYPE NOT SET

Explanation: The setting for the field type to be used was not recognized. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ22033E SVC 99 INFORMATION RETRIEVAL FAILED; DDNAME: ddname R15= nnnn S99INFO= X'nnnn' S99ERROR= X'nnnn'

Explanation: The SVC 99 information retrieval of the data set name of the identified DD has failed. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ22034E THE DATA SET NAME OF DDNAME: DBD01 DOES NOT HAVE DB2 FORMAT; datasetname

Explanation: The data set name of the data set allocated to DD DBD01 does not have a DB2 data set name form. The directory information being updated by DB2UPDATE has been determined to no longer reside in the DBD01 table space. DB2UPDATE is attempting to dynamically allocate the SYSDBDXA table space where this data should reside but cannot determine what the data set name should be. Processing terminates.

User response: Add a SYSDBDXA DD that points to the SYSDBDXA table space or point the DBD01 DD to a DBD01 table space that has a DB2 format data set name.

CKZ22035I NO DBD INFORMATION WAS FOUND IN DBD01; WILL DYNAMICALLY ALLOCATE SYSDBDXA TO USE

Explanation: No DBD information was found in DBD01. The directory information being updated by DB2UPDATE has been determined to no longer reside in the DBD01 table space. DB2UPDATE will attempt to dynamically allocate the SYSDBDXA table space, where this data should reside, to use.

User response: No action is required. A SYSDBDXA DD that points to the SYSDBDXA table space can be added to the JCL.

CKZ22041E CKZ00900 UNEXPECTED RESULTS; error text LOC=IIIII

Explanation: An unexpected condition occurred calling program CKZ00900. 'error text' has a description of the problem. IIIII is the location where the error occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ22042E REQUIRED INI SECTION/TOKEN MISSING: SECTION=section TOKEN=token | REQUIRED INI VALUE MISSING FOR SECTION=section TOKEN=token | INVALID INI VALUE FOR SECTION=section TOKEN=token error text for invalid value

Explanation: An error occurred validating the CKZINI member options. Processing terminates.

User response: Correct the CKZINI member.

CKZ22050E ERROR IN PARAMETERS FOR keyword

Explanation: The parameters for the indicated keyword were incorrect. Processing terminates.

User response: Check the keyword parameters. Mutually exclusive keywords may have been used.

CKZ22051E REQUIRED KEYWORD MISSING: keyword

Explanation: A keyword required for processing has been omitted. Processing terminates.

User response: Specify the required keyword.

CKZ22053E MAXIMUM LENGTH EXCEEDED FOR KEYWORD: keyword

Explanation: The operand entered for a keyword exceeded the maximum length allowed for the operand. Processing terminates.

User response: Correct the length of the keyword's operand.

CKZ22054E KEYWORD HAS MORE THAN 1 OPERAND; ONLY 1 ALLOWED: keyword | KEYWORD HAS MORE THAN 2 OPERANDS; ONLY 2 ALLOWED: keyword | KEYWORD HAS MORE THAN 8 OPERANDS;

ONLY 8 ALLOWED: keyword

Explanation: More operands were specified for a keyword than are permitted. Processing terminates.

User response: Correct the keyword.

CKZ22056E NOTHING SPECIFIED FOR
KEYWORD: keyword

Explanation: A keyword was entered without an appropriate operand. Processing terminates.

User response: Specify an appropriate operand for the keyword.

CKZ22057E DUPLICATE FOUND; KEYWORD:
keyword ENTRY: entry

Explanation: The indicated 'entry' for the keyword was previously specified. Processing terminates.

User response: Remove the duplicate entry.

CKZ22058E INVALID VALUE IN KEYWORD:
keyword VALUE: value error text

Explanation: The value in the keyword is invalid. 'error text' indicates the problem detected with the value. Processing terminates.

User response: Correct the value specified in the keyword.

CKZ22060E TARGET STORAGE GROUP NAME
MUST BE THE SAME LENGTH AS
THE SOURCE STORAGE GROUP
NAME | TARGET STORAGE GROUP
NAME: target-sgname | SOURCE
STORAGE GROUP NAME:
source-sgname

Explanation: The target storage group name is not the same length as the source storage group name. For DB2 Version 8, the names must be the same length. Processing terminates.

User response: Correct the values specified so they are the same length.

CKZ22068E UNPAIRED ENTRY IN KEYWORD:
keyword

Explanation: For DB2-HLQS, there must be a source alias followed by a target alias. For DB2-MEMBERS, there must be a source member name followed by a target member name. For STOGROUPS, there must be a source name followed by a target name. An uneven number of entries was specified. Processing terminates.

User response: Correct the keyword specification.

CKZ22069E THE USE OF keyword1 REQUIRES
THE USE OF keyword2

Explanation: Keyword1 was specified. Its use requires that keyword2 must also be specified. Processing terminates. An uneven number of entries was specified. Processing terminates.

User response: Correct the keyword specification.

CKZ22074E NAME USED AS BOTH SOURCE AND
TARGET: name IN KEYWORD:
keyword

Explanation: A target entry was also specified as a source in the indicated keyword. Processing terminates.

User response: Correct the keyword specification.

CKZ22075E NAME USED AS TARGET MULTIPLE
TIMES: name IN KEYWORD: keyword

Explanation: A target entry was specified for multiple sources in the indicated keyword. Processing terminates.

User response: Correct the keyword specification.

CKZ22085I PAIRS FOR KEYWORD: keyword

Explanation: The names indicated for the keyword have been accepted for processing.

User response: No action is required.

CKZ22086I VALIDATING KEYWORD: keyword

Explanation: Parsing is checking the indicated keyword indicated in the command.

User response: No action is required.

CKZ22101I hh:mm:ss BSDS UPDATING STARTED
- PROGRAM REV=rrr (**
SIMULATION **) | hh:mm:ss BSDS
UPDATING COMPLETED; RETURN
CODE=nnn

Explanation: DB2UPDATE processing message.

User response: No action is required.

CKZ22106E ERROR CALLING CKZ01VV1 tttttt
FUNCTION: function R15=nnnn
R0=nnnnnnnn LOC=lllll

Explanation: A problem occurred using a dataspace. tttttt is the name of the internal table. lllll is the location where the error occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

**CKZ22107W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn**

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

**CKZ22109E ERROR ACCESSING BSDS FILE;
LOC=IIIII | ERROR ACCESSING
JOURNAL; LOC=IIIII**

Explanation: A VSAM error occurred accessing a file. Processing terminates.

User response: See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ22110E DUPLICATE JOURNAL ENTRY;
LOC=IIIII**

Explanation: A duplicate record was detected. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ22111E BSDS DATA SHARING RECORD(S)
NOT FOUND**

Explanation: An expected record was not found in the BSDS file. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ22112E JOURNAL DB2 CONTROL RECORD
NOT FOUND | JOURNAL DB2
CONTROL RECORD IS WRONG
VERSION**

Explanation: The journal DB2 control record as not found or was not match the expected format. If the format is the problem, the record is printed. Processing terminates.

User response: Verify that different releases of DB2 Cloning Tool have not been run using the same journal data set. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains this message.

**CKZ22113E UNKNOWN RECORD TYPE READ
FROM: ttttttt; LOC=IIIII**

Explanation: An unexpected record was found in the dataspace. The record is printed. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ22130I PROCESSING BSDSnn

Explanation: The indicated BSDS is being processed.

User response: No action is required.

**CKZ22135E BSDS CONTROL RECORD NOT
FOUND**

Explanation: The control record was not found in the BSDS. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ22136W BSDS DB2 CATALOG DOES NOT
MATCH KEYWORD**

Explanation: The DB2 catalog alias found in the BSDS does not match a source name specified in the DB2-HLQS keyword. Processing continues.

User response: Correct the DB2-HLQS name(s) specified if they are in error. If DB2UPDATE is being rerun and has already successfully updated the BSDS, the message may be ignored.

**CKZ22137E BSDS RECORD NOT FOUND FOR
recordtype**

Explanation: An expected BSDS record type was not found. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ22138W BSDS ACTIVE LOG # DSN: dsname
DOES NOT MATCH RENAME MASKS**

Explanation: An active log data set name does not match the rename masks specified during DB2 Cloning Tool RENAME. Processing continues.

User response: The active log data set(s) should be on the volumes copied by DB2 Cloning Tool COPY and the old/new names specified in DB2 Cloning Tool RENAME. If DB2UPDATE is being rerun and has already successfully updated the BSDS, the message may be ignored.

**CKZ22139W BSDS ARCHIVE LOG # DSN: dsname
VOL: volser errortext**

Explanation: Either the archive log data set name does not match the rename masks specified during RENAME, or, the archive log was not on a volume copied by DB2 Cloning Tool COPY. Processing continues.

User response: No action is required.

CKZ22140I BSDS UPDATED RECORD FOR
recordtype

Explanation: A simulation of DB2UPDATE was requested. The updated BSDS record is printed, but, not rewritten.

User response: No action is required.

CKZ22141W BSDS DSN: datasetname DOES NOT
MATCH RENAME MASKS

Explanation: A BSDS data set name in the BSDS was not renamed due to there being no RENAME-MASKS entry that matches it.

User response: Correct the RENAME-MASKS specification for DB2 Cloning Tool RENAME if they are in error.

CKZ22143E UNABLE TO RENAME
DSN=datasetname USING
MASK=target rename mask

Explanation: The new name of a data set will exceed 44 characters. Processing terminates.

User response: Correct the RENAME-MASKS specification for DB2 Cloning Tool RENAME.

CKZ22144E ERROR DETERMINING TARGET
VOLUME SERIAL LENGTH; SOURCE:
volser

Explanation: The program was unable to determine the length of the target volume serial. The volume pair entry is printed. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ22150W THE BSDS HAS A DDF RECORD BUT
THE DDF KEYWORD WAS NOT
SPECIFIED

Explanation: The BSDS contains a DDF record but the DDF keyword was not specified. Processing continues.

User response: Add a DDF keyword if the DDF values should be changed.

CKZ22151W BSDS DB2 CATALOG MATCHES
TARGET VALUE

Explanation: The DB2 catalog alias found in the BSDS matches a target name specified in the DB2-HLQS keyword. Processing continues.

User response: Correct the DB2-HLQS name(s) specified if they are in error. If DB2UPDATE is being rerun and has already successfully updated the BSDS, the message may be ignored.

CKZ22152W MEMBER: membername DOES NOT
MATCH DB2-MEMBERS MASKS

Explanation: A member name in the BSDS was not renamed due to there being no DB2-MEMBERS entry that matches it. Processing continues.

User response: Correct the DB2-MEMBERS name(s) specified if they are in error.

CKZ22153W THE BSDS IS DATA SHARING BUT
THE DB2-GROUP AND
DB2-MEMBERS KEYWORDS WERE
NOT SPECIFIED

Explanation: The BSDS for this DB2 system is enabled for data sharing but the data sharing related keywords DB2-GROUP and DB2-MEMBERS were not specified. Processing continues.

User response: Add the DB2-GROUP and DB2-MEMBERS keywords.

CKZ22154W THE BSDS IS NOT DATA SHARING
BUT THE DB2-GROUP AND
DB2-MEMBERS KEYWORDS WERE
SPECIFIED

Explanation: The BSDS for this DB2 system is not enabled for data sharing but the data sharing related keywords DB2-GROUP and DB2-MEMBERS were specified. Processing continues.

User response: Remove the DB2-GROUP and DB2-MEMBERS keywords.

CKZ22155W THE DDF ALIAS KEYWORD WAS
SPECIFIED BUT THE DDF RECORD IS
NOT V8 FORMAT

Explanation: The ALIAS keyword was included in the DDF keyword but the DDF record in the BSDS indicates it is for a release prior to DB2 Version 8 that does not support ALIAS. Processing continues.

User response: Remove the ALIAS keyword.

CKZ22156W THE DDF keyword KEYWORD WAS
SPECIFIED BUT THE DDF RECORD IS
NOT V9 FORMAT

Explanation: The keyword was included in the DDF keyword but the DDF record in the BSDS indicates it is for a release prior to DB2 Version 9.1 that does not support the keyword. Processing continues.

User response: Remove the keyword.

CKZ22157W THE DDF ALIAS SECPORT VALUE WAS SPECIFIED BUT THE DDF RECORD IS NOT V9 FORMAT

Explanation: The ALIAS secpport value was included in the DDF keyword but the DDF record in the BSDS indicates it is for a release prior to DB2 Version 9.1 that does not support it Processing continues.

User response: Remove the secpport value.

CKZ22158I A BACKUP SYSTEM RECORD IS IN THE BSDS BUT IT HAS NO ENTRIES

Explanation: A Backup System record was found in the BSDS and it contained no entries.

User response: No action is required.

CKZ22160E SVC 99 INFO RETRIEVAL FAILED; DDNAME=ddname ERROR=xxxx INFO=xxxx

Explanation: A problem occurred using SVC 99 information retrieval to get data set information about the data set allocated to DD ddname. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

CKZ22201I hh:mm:ss LINEAR FILE UPDATING STARTED - PROGRAM REV=rrr (SIMULATION **) | hh:mm:ss LINEAR FILE UPDATING COMPLETED; RETURN CODE=nnn RECORD COUNT=nnn**

Explanation: DB2UPDATE processing message.

User response: No action is required.

CKZ22204E OPEN FAILED FOR DDNAME=ddname

Explanation: 'ddname' was allocated for DB2 Cloning Tool to use, but, the open for the file failed. Processing terminates.

User response: If unable to determine the reason the open failed, contact IBM Software Support. Have available the listing that contains this message.

CKZ22207W ERROR CALLING CKZ01HEX; FUNCTION: function R15=nnnn

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

CKZ22210E ERROR ACCESSING LINEAR FILE; DDNAME=ddname R15=nnnn ERROR=nnn FUNCTION=function LOC=lllll RBA OF RECORD: nnnnnnnnnnnn X'nnnnnnnn'

Explanation: A VSAM error occurred accessing the indicated file. Processing terminates.

User response: If unable to resolve problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ22212E UNABLE TO ESTABLISH ESTAEX; R15=nnnn

Explanation: The program was not able to establish an estaex environment. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ22230I PROCESSING DDNAME: ddname

Explanation: The indicated linear file is being processed.

User response: No action is required.

CKZ22231I THE FILE'S ENDING RBA IS: nnnnnnnnnnnn X'nnnnnnnn'

Explanation: The ending rba of the linear file being processed.

User response: No action is required.

CKZ22232I NO DBD INFORMATION FOUND

Explanation: No DBD information was found. This can happen when the DBD01 table space is from a DB2 10 system and the DBD information has been moved to the SYSDBDXA table space.

User response: No action is required.

CKZ22240I UPDATED, RECORD NUMBER: rrr FIELDS CHANGED: fff | CHILD RECORD UPDATED

Explanation: A simulation of DB2UPDATE was requested. A record in the indicated file matched on DB2-HLQS, STOGROUPS, and/or volume serial number. The record is not rewritten. rrr is the relative record number in the file. fff is the number of occurrences of a field that was changed. 'CHILD RECORD UPDATED' indicates that the record had other entries associated with it.

User response: No action is required.

CKZ22241E RECORD FLAGGED FOR CHANGE, BUT, NO CHANGE WAS SUCCESSFUL

Explanation: The initial check indicated a record needed to be updated, but, the subsequent check did not find anything to update. Should not occur. The record is printed. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ22242I ddname CHANGED RECORDS: nnnn CHANGED FIELDS: ffff

Explanation: The linear file had nnnn records updated; ffff fields in those records were modified.

User response: No action is required.

CKZ22243W ddname INCONSISTENT RECORD SKIPPED: RECORD NUMBER: nnnn

Explanation: An entry in the indicated file had was flagged as inconsistent. No changes will be made to this record.

User response: No action is required.

CKZ22244E ddname UNKNOWN SEGMENT; RECORD NUMBER: nnnn

Explanation: A record part in the indicated file was not recognized. The first 16 bytes of recognized segments are printed; all of the unknown segment is printed. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ22245W ddname DBD IS NOT TYPE 2039; RECORD NUMBER: nnnn

Explanation: A record part in the indicated file was set as a DBD. The type should be 2039 but is not. The unknown DBD is printed. Processing continues.

User response: No action is required.

CKZ22246E ddname ERROR PARSING RECORD NUMBER: nnnn

Explanation: The ESTAEX routine was entered while processing the indicated record. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ22247E INVALID RBA CALCULATED: nnnnnnnnnnnn X'nnnnnnnn'

Explanation: The RBA calculated for a required record is greater than the end RBA for the file. The record that

caused the RBA calculation is printed. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ22250W VCAT NAME vcatname NOT CHANGED, IT DOES NOT MATCH KEYWORD

Explanation: A VCAT found in DBD01 does not match a source name specified in the DB2-HLQS keyword. Processing continues.

User response: Correct the DB2-HLQS name(s) specified if they are in error.

CKZ22260E TARGET STORAGE GROUP NAME MUST BE THE SAME LENGTH AS THE SOURCE STORAGE GROUP NAME | TARGET STORAGE GROUP NAME: target-sgname | SOURCE STORAGE GROUP NAME: source-sgname

Explanation: The target storage group name is not the same length as the source storage group name. For DB2 Version 8, the names must be the same length. Processing terminates.

User response: Correct the values specified so they are the same length.

CKZ22261W STORAGE GROUP NAME TOO LONG storage-group-name

Explanation: The storage groupname found in DBD01 is longer than 30 characters. Processing continues.

User response: This storage group name must be changed manually by use of the appropriate DB2 statements.

CKZ22299E ABEND DURING LINEAR FILE UPDATE

Explanation: An abend occurred during the DBD01 update. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ22301I hh:mm:ss DB2 XCF CLEAN UP STARTED - PROGRAM REV=rrr (SIMULATION **) | hh:mm:ss DB2 XCF CLEAN UP COMPLETED; RETURN CODE=nnn**

Explanation: DB2UPDATE processing message.

User response: No action is required.

**CKZ22307W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn**

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

**CKZ22310E UNEXPECTED RECORD TYPE
ENCOUNTERED: type LOC=IIIII**

Explanation: An unexpected condition was encountered while processing an XCF structure list.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ22311E ERROR CALLING IXCQUERY; RC: rrr
RSN: ssssssss LOC=IIIII | ERROR
CALLING IXLFORCE; RC: rrr RSN:
ssssssss LOC=IIIII**

Explanation: An error was received from the system macro call.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ22312E ALLOCATED STRUCTURE TABLE IS
FULL | CONNECTED STRUCTURE
TABLE IS FULL | BAD STATUS
STRUCTURE TABLE IS FULL**

Explanation: An internal table is too small.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ22313E ERROR ACCESSING JOURNAL FILE;
LOC=IIIII**

Explanation: A VSAM error occurred accessing the journal file. Processing terminates.

User response: See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ22314E JOURNAL DB2 MEMBER RECORD
NOT FOUND**

Explanation: The journal DB2 member record was not found. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ22315E JOURNAL DB2 MEMBER RECORD IS
WRONG VERSION**

Explanation: The journal DB2 control record does not match the expected format. The record is printed. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ22320I XCF STRUCTURES FOR DB2 DATA
SHARING GROUP groupname: list of
structures**

Explanation: This is a list of the DB2 XCF structures for the groupname.

User response: No action is required.

**CKZ22321E THE FOLLOWING XCF STRUCTURES
HAVE AN UNDESIRABLE STATUS: list
of structures**

Explanation: This is a list of the DB2 XCF structures that have a status that will not allow the clean up of the structures.

User response: Verify that there are no target DB2 systems running.

**CKZ22322I THE FOLLOWING XCF STRUCTURES
WILL HAVE FAILED CONNECTIONS
REMOVED: | THE FOLLOWING XCF
STRUCTURES SHOULD HAVE FAILED
CONNECTIONS REMOVED: list of
structures**

Explanation: This is a list of the DB2 XCF structures that have failed connections that, (1) will be removed, or (2) should be removed.

User response: (1) None. (2) The list of structures should have their failed connections removed manually.

**CKZ22323I THE FOLLOWING XCF STRUCTURES
WILL BE DEALLOCATED: | THE
FOLLOWING XCF STRUCTURES
SHOULD BE DEALLOCATED: list of
structures**

Explanation: This is a list of the DB2 XCF structures that, (1) will be deallocated, or (2) should be deallocated.

User response: (1) None. (2) The list of structures should be manually deallocated.

**CKZ22331I XCF STRUCTURES NOT CLEANED UP
DUE TO ERRORS**

Explanation: Due to errors all the DB2 XCF structures have not been cleaned up. The specific errors are described in other messages.

User response: No action is required.

CKZ22332I ALL XCF STRUCTURES HAVE THE PROPER STATUS, NO CLEAN UP NECESSARY

Explanation: The DB2 XCF structures are all in 'NOT ALLOCATED' status. There is no clean up necessary.

User response: No action is required.

CKZ22333W NO XCF STRUCTURES FOUND FOR THIS DB2 SHARING GROUP

Explanation: There were no XCF structures found for the DB2 sharing group.

User response: No action is required.

CKZ22334I XCF STRUCTURES CLEANED UP

Explanation: The XCF structures for the DB2 sharing group have been cleaned up.

User response: No action is required.

CKZ22335I XCF STRUCTURES NOT CLEANED UP DUE TO SIMULATION

Explanation: A simulation of DB2UPDATE was requested. The DB2 XCF structures were not cleaned up.

User response: No action is required.

CKZ22336W XCF STRUCTURES SHOULD BE CLEANED UP BEFORE STARTING TARGET DB2

Explanation: The DB2 XCF structures for the DB2 sharing group should be cleaned up prior to starting the target DB2.

User response: Clean up the target DB2 XCF structures prior to starting the target DB2.

CKZ22337I XCF STRUCTURES CAN NOT BE CLEANED UP DUE TO ERRORS

Explanation: A simulation of DB2UPDATE was requested. The DB2 XCF structures cannot be cleaned up due to errors. The specific errors are described in other messages.

User response: Clean up the target DB2 XCF structures prior to starting the target DB2.

CKZ22340W XCF STRUCTURE DOES NOT HAVE EXPECTED DB2 STRUCTURE NAME,BYPASSED

Explanation: The XCF structure name has a prefix of the target DB2 group name but does not follow the

naming conventions for a DB2 XCF structure. The structure is bypassed from further processing.

User response: No action is required.

CKZ22341W IXLFORCE REQUEST IS PENDING, THE SYSTEM WILL PROCESS IT WHEN THE DELAY CAUSE IS RESOLVED

Explanation: The IXLFORCE request could not be processed immediately. The request is pending, and the system will process it when the condition causing the delay is resolved.

User response: No action is required.

CKZ22342E USER DOES NOT HAVE PROPER SAF AUTHORIZATION TO CLEAN UP THE XCF STRUCTURE

Explanation: The user running the job does not have SAF (RACF) authority to clean up the DB2 XCF structures.

User response: Give the user running the job the proper authorization to allow clean up of the structures. Authorization is by entities with a format of IXLSTR.structurename in the FACILITY class.

CKZ22350I XCF MEMBERS FOR DB2 DATA SHARING GROUP groupname: list of members

Explanation: A list of the members found for the groupname.

User response: No action is required.

CKZ22351I THE FOLLOWING XCF MEMBERS HAVE AN UNDESIRABLE STATUS: list of members

Explanation: The indicated members do not have quiesced or failed status.

User response: The target DB2 system needs to be shut down if it is active.

CKZ22351W THE FOLLOWING XCF MEMBERS HAVE AN UNDESIRABLE STATUS: list of members

Explanation: The indicated members do not have quiesced or failed status.

User response: The target DB2 system needs to be shut down if it is active.

CKZ22352I NO XCF MEMBERS FOUND FOR DB2 DATA SHARING GROUP *groupname*, NO CLEAN UP NECESSARY

Explanation: No DB2 XCF members were found for the *groupname*.

User response: No action is required.

CKZ22354I XCF MEMBERS CLEANED UP

Explanation: The DB2 XCF members have been cleaned up.

User response: No action is required.

CKZ22355I XCF MEMBERS NOT CLEANED UP DUE TO *reason*

Explanation: The DB2 XCF members have not been cleaned up. Reason indicates why the cleanup did not happen; SIMULATION if SIMULATE was specified, or DB2-XCFCLEAN(N) if DB2-XCFCLEAN(N) was specified.

User response: No action is required.

CKZ22357I XCF MEMBERS NOT CLEANED UP DUE TO ERRORS

Explanation: The DB2 XCF members have not been cleaned up. Other messages will describe the problem encountered.

User response: Check for other messages indicating the error which prevented the clean up.

CKZ22358W THIS DB2 MEMBER (*membername*) CAN NOT BE STARTED DUE TO XCF MEMBERID MISMATCH

Explanation: The start up of the target DB2 member will fail due to the memberid in the BSDS being different than the memberid in the XCF group member.

User response: Do not start up the DB2 member.

CKZ22359W XCF MEMBER *membername* NOT CLEANED UP DUE TO ITS UNDESIRABLE STATUS

Explanation: The XCF member *membername* was not cleaned up because it did not have quiesced or failed status. XCF members will only be cleaned up if they have quiesced or failed status.

User response: The target DB2 system needs to be shut down if it is active. 25957

CKZ22401I *hh:mm:ss* DIRECTORY UPDATING STARTED - PROGRAM REV=*rrr* (SIMULATION **) | *hh:mm:ss* DIRECTORY UPDATING COMPLETED; RETURN CODE=*nnn***

Explanation: DB2UPDATE processing message.

User response: No action is required.

| **CKZ22403I DDNAME=*dd-name* ALLOCATED FOR DSN=*data-set-name***

| **Explanation:** *ddname* has been dynamically allocated for the specified data set.

| **User response:** No action is required.

CKZ22404E OPEN FAILED FOR DDNAME=*ddname*

Explanation: '*ddname*' was allocated for DB2 CLONING TOOL to use, but, the open for the file failed. Processing terminates.

User response: If unable to determine the reason the open failed, contact IBM Software Support. Have available the listing that contains this message.

| **CKZ22405E ALLOCATION FAILED FOR DSN: *data-set-name***

| **Explanation:** Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

| **User response:** If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

| **CKZ22405W DEALLOCATION FAILED FOR DDNAME: *ddname***

| **Explanation:** Dynamic deallocation for a *ddname* failed. The associated z/OS messages are displayed. Processing continues.

| **User response:** If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ22407W ERROR CALLING CKZ01HEX; FUNCTION: *function* R15=*nnnn*

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

CKZ22410E ERROR ACCESSING LINEAR FILE;
DDNAME=ddname R15=nnnn
ERROR=nnn FUNCTION=function
LOC=IIIII RBA OF RECORD:
X'nnnnnnnn_nnnnnnnn' PAGE
NUMBER X'nnnnnnnn'

Explanation: A VSAM error occurred accessing the indicated file. Processing terminates.

User response: If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ22412E UNABLE TO ESTABLISH ESTAEX;
R15=nnnn

Explanation: The program was not able to establish an estaex environment. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ22423E SVC 99 INFORMATION RETRIEVAL
FAILED; DDNAME: ddname R15= nnnn
S99INFO= X'nnnn' S99ERROR= X'nnnn'

Explanation: The SVC 99 information retrieval of the data set name of the identified DD has failed. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains these messages.

CKZ22424E THE DATA SET NAME OF DDNAME:
ddname DOES NOT HAVE DB2
FORMAT; data-set-name

Explanation: The data set name of the data set allocated to the indicated ddname does not have a DB2 data set name form. DB2UPDATE is trying to determine if there are multiple data sets for the DB2 directory LOB space SYSDBDXA that need to be processed, but cannot determine what the data set names should be. Processing terminates.

User response: Change the JCL statement for the indicated ddname to point to the DB2 directory LOB space SYSDBDXA, which has a DB2 format data set name and also has a low level qualifier of A001.

CKZ22425E THE NUMBER OF DIRECTORY DATA
SETS DISCOVERED IS GREATER
THAN THE EXPECTED MAXIMUM OF
254

Explanation: More than 254 DB2 directory LOB space SYSDBDXA data sets have been discovered. A maximum of 254 data sets is expected. Processing terminates.

User response: Verify the data set being used corresponds to the correct DB2 directory LOB space

| SYSDBDX data set. If unable to resolve problem,
 | contact IBM Software Support. Have available the
 | listing that contains this message.

CKZ22430I PROCESSING DDNAME: ddname

Explanation: The indicated file is being processed.

User response: No action is required.

CKZ22431I THE ENDING RBA FOR DDNAME:
ddname IS: X'nnnnnnnnnn_nnnnnnnnn'

| **Explanation:** The ending RBA of the directory file
 | being processed.

| **User response:** No action is required.

CKZ22432E END OF FILE ENCOUNTERED WHILE
READING HEADER PAGE

Explanation: End of file was encountered while reading the header page. The data set allocated to the SYSDBDXA DD appears to have no data in it. Processing terminates.

User response: Ensure the correct data set is allocated to the SYSDBDXA DD. If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ22433E FIRST PAGE IS NOT LOB HEADER
PAGE

Explanation: The first page of the data set allocated to the SYSDBDXA DD is not a LOB header page. Processing terminates.

User response: Ensure the correct data set is allocated to the SYSDBDXA DD. If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ22434E INTERNAL ERROR; LOC=IIIII reason
text

Explanation: An internal error has occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ22435E UNEXPECTED PAGE TYPE
RETURNED FROM READ DIRECT

Explanation: Read direct got a page type that was not a LOB map or LOB header. The page is printed. Processing terminates.

User response: Ensure the correct data set is allocated to the SYSDBDXA DD. If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ22436W UNEXPECTED DBD IDENTIFIER; MAP PAGE NUMBER: X'nnnnnnnn'

Explanation: The identifier of a DBD is not as expected. The DBD is skipped. The DBD is printed. Processing continues.

User response: No action is required.

CKZ22437W UNABLE TO UPDATE DBD DUE TO POSSIBLY UNWRITTEN PAGES; MAP PAGE NUMBER: X'nnnnnnnn' DATA PAGE NUMBER: X'nnnnnnnn' DB2UPDATE WILL NEED TO RUN AGAIN AFTER THE TARGET DB2 IS STARTED AND RESOLVES THE UNWRITTEN PAGES

Explanation: A page that is part of a DBD appears to be empty or has old data in it. This implies that possibly not all the pages of the DBD had been externalized by DB2 when the volumes were cloned. The updating of this DBD is bypassed. Processing continues.

User response: DB2UPDATE with the DBD01ONLY keyword will need to be run after the target DB2 is started, DB2FIX DATABASES(DB2) has been run, and the target DB2 stopped. DB2FIX DATABASES(DB2) is also expected to issue message CKZ23526W when this happens.

CKZ22440I UPDATED, RECORD STARTING ON PAGE: X'nnnnnnnn' FIELDS CHANGED: fff

Explanation: A simulation of DB2UPDATE was requested. A record in the file matched on DB2-HLQS, STOGROUPS, and/or volume serial number. The record is not rewritten. nnnnnnnn is the hex page number in the file. fff is the number of occurrences of a field that was changed.

User response: No action is required.

CKZ22442I ddname CHANGED RECORDS: nnnn CHANGED FIELDS: ffff

Explanation: The directory file had nnnn records updated; ffff fields in those records were modified.

User response: No action is required.

CKZ22443W ddname INCONSISTENT PAGE; PAGE NUMBER: X'nnnnnnnn' LOC=IIIII

Explanation: A page in the directory file was flagged as inconsistent. Processing continues.

User response: No action is required.

CKZ22444E PAGE READ IS NOT EXPECTED PAGE NUMBER; REQUESTED: X'nnnnnnnn' READ: x'nnnnnnnn' LOC=IIIII

Explanation: A direct read of a page did not return the requested page. The page is printed. Processing terminates.

User response: Ensure the correct data set is allocated to the SYSDBDXA DD. If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ22445E CHECK BYTE ON PAGE IS NOT AS EXPECTED; EXPECTED: X'nn' GOT x'nn' LOC=IIIII

Explanation: The check byte on a page does not have the expected value. The page is printed. Processing terminates.

User response: Ensure the correct data set is allocated to the SYSDBDXA DD. If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ22446E ddname ERROR PARSING RECORD AT RBA: X'nnnnnnnn_nnnnnnnn'

Explanation: The ESTAEX routine was entered while processing the indicated record. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ22450W VCAT NAME vcatname NOT CHANGED, IT DOES NOT MATCH KEYWORD

Explanation: A VCAT found in the file does not match a source name specified in the DB2-HLQS keyword. Processing continues.

User response: Correct the DB2-HLQS name(s) specified if they are in error.

CKZ22460E TARGET STORAGE GROUP NAME MUST BE THE SAME LENGTH AS THE SOURCE STORAGE GROUP NAME | TARGET STORAGE GROUP NAME: target-sgname | SOURCE STORAGE GROUP NAME: source-sgname

Explanation: The target storage group name is not the same length as the source storage group name. For DB2 Version 8 or above, the names must be the same length. Processing terminates.

User response: Correct the values specified so they are the same length.

CKZ22461W STORAGE GROUP NAME TOO LONG
storage-group-name

Explanation: The storage group name found in the directory is longer than 30 characters. Processing continues.

User response: This storage group name must be changed manually by use of the appropriate DB2 statements.

CKZ22499E ABEND DURING DIRECTORY UPDATE

Explanation: An abend occurred during the directory update. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ23001I hh:mm:ss DB2RESUME STARTED - PROGRAM REV= | hh:mm:ss DB2RESUME COMPLETED; RETURN CODE=nn RECORD COUNT=nnn

Explanation: DB2RESUME command processing message.

User response: No action is required.

CKZ23003I DDNAME=ddname ALLOCATED FOR DSN=datasetname

Explanation: 'ddname' has been dynamically allocated for the indicated data set.

User response: No action is required.

CKZ23004E DDNAME MISSING: ddname

Explanation: 'ddname' was specified for DB2 Cloning Tool to use. Processing terminates.

User response: Either correct the ddname specified, or add the appropriate ddname to the job's JCL.

CKZ23005E ALLOCATION FAILED FOR DSN: datasetname

Explanation: Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ23005W DEALLOCATION FAILED FOR DDNAME: ddname

Explanation: Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ23007W ERROR CALLING CKZ01HEX; FUNCTION: function R15=nnnn

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

CKZ23011E JOURNAL CONTROL RECORD NOT FOUND

Explanation: An expected record was not found in the DB2 Cloning Tool journal file. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ23015E THE COPY PROCESS DID NOT COMPLETE SUCCESSFULLY

Explanation: The journal indicates that the COPY command did not complete successfully. Processing terminates.

User response: Check that the COPY and the RENAME commands have completed successfully before initiating this command.

CKZ23016E THE RENAME PROCESS DID NOT COMPLETE SUCCESSFULLY; FAILED TASK=failed rename task

Explanation: The journal indicates that the RENAME command did not complete successfully. Processing terminates.

User response: Check that the COPY and the RENAME commands have completed successfully before initiating this command.

CKZ23017W NO RECORDS WERE PROCESSED FROM DB2RECS FILE

Explanation: The DB2RECS file is empty.

User response: Check that the COPY and the RENAME commands have completed successfully before initiating this command.

CKZ23031W PRIOR STEP WAS A SIMULATION

Explanation: The journal indicates that the COPY command or the RENAME command was a simulation. No DB2 recatalog will be done.

User response: No action is required.

**CKZ23045E ERROR ACCESSING BCS=bcs dsname;
LOC=IIIII**

Explanation: A VSAM error occurred accessing the indicated BCS. Processing terminates.

User response: See associated CKZERRnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ23047E DUPLICATE BCS ENTRY

Explanation: A duplicate record was detected. Processing terminates.

User response: This is not expected in DB2RESUME. Contact IBM Software Support. Have available the listing that contains this message.

**CKZ23048I OPTIONS IN EFFECT FOR THIS
EXECUTION: merged options**

Explanation: Informational message indicating how DB2RESUME will handle the options. The displayed options are derived from the CKZINI and any overriding specifications in the command input.

User response: No action is required.

**CKZ23051E REQUIRED KEYWORD MISSING:
keyword**

Explanation: A keyword required for processing has been omitted. Processing terminates.

User response: Specify the required keyword.

**CKZ23052E REQUIRED INI SECTION/TOKEN
MISSING: SECTION=section
TOKEN=token | REQUIRED INI
VALUE MISSING FOR
SECTION=section TOKEN=token |
INVALID INI VALUE FOR
SECTION=section TOKEN=token**

Explanation: An error occurred validating the CKZINI member options. Processing terminates.

User response: Correct the CKZINI member.

**CKZ23053E KEYWORD: keyword MAXIMUM
LENGTH: nnn EXCEEDED**

Explanation: The operand entered for a keyword exceeded the maximum length allowed for the operand. nnn is the maximum allowed length for the keyword. Processing terminates.

User response: Correct the length of the keyword's operand.

**CKZ23054E KEYWORD HAS MORE THAN 1
OPERAND; ONLY 1 ALLOWED:
keyword**

Explanation: Multiple operands were detected for a keyword; only one operand is permitted. Processing terminates.

User response: Correct the keyword to use one operand.

**CKZ23056E NOTHING SPECIFIED FOR
KEYWORD: keyword**

Explanation: A keyword was entered without an appropriate operand. Processing terminates.

User response: Specify an appropriate operand for the keyword.

**CKZ23058E INVALID VALUE IN KEYWORD:
keyword VALUE: value error text**

Explanation: The value in the keyword is invalid. 'error text' indicates the problem detected with the value. Processing terminates.

User response: Correct the value specified in the keyword.

**CKZ23060I WAITING FOR EXCLUSIVE CONTROL
OF BCS bcs name**

Explanation: The BCS is currently in use by another job. The wait will continue until the BCS is no longer in use by another job or the wait time limit is exceeded. Processing continues.

User response: No action is required.

**CKZ23061E UNABLE TO ALLOCATE BCS: bcs
name; WAIT TIME LIMIT EXCEEDED**

Explanation: The wait for exclusive control of the BCS has exceeded the wait time limit. Processing terminates.

User response: Change the scheduling of the jobs so the DB2 Cloning Tool job does not run when another job has the BCS allocated. Or increase the wait time limit so the DB2 Cloning Tool job can wait longer for the other job to terminate. The wait time limit is set by the CKZINI parameter

CONCURRENT_EXECUTIONS_WAIT_TIME.

CKZ23501I hh:mm:ss DB2FIX STARTED -
PROGRAM REV=rrr | hh:mm:ss
DB2FIX COMPLETED; RETURN
CODE=nnn

Explanation: DB2FIX processing message.

User response: No action is required.

CKZ23506E ERROR CALLING CKZ01VV1 tttttt
FUNCTION: function R15=nnnn
R0=nnnnnnnn LOC=llll

Explanation: A problem occurred using a dataspace. tttttt is the name of the internal table. llll is the location where the error occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

CKZ23507W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

CKZ23508E UNABLE TO LOAD PROGRAM:
program name

Explanation: The indicated program name was not found. Processing terminates.

User response: Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

CKZ23520E DSNALI ERROR; FUNCTION=
function RC=nnnn RSN=nnnnnnnn

Explanation: An error occurred calling DSNALI. Function is the CAF function requested and RSN is the DB2 error reason. Processing terminates.

User response: Check the DB2 Messages and Codes manual for the RSN value. If unable to resolve the problem, contact IBM Software Support.

CKZ23521E IFI ERROR; FUNCTION= function
RC=nnnn RSN=nnnnnnnn

Explanation: An error occurred using the DB2 IFI interface. Function is the IFI function requested and RSN is the DB2 error reason. Processing terminates.

User response: Check the DB2 Messages and Codes manual for the RSN value. If unable to resolve the

problem, contact IBM Software Support.

CKZ23522E DB2 SUBSYSTEM: db2ssid NOT
ACTIVE, RSN=00F30002 | DB2
SUBSYSTEM: db2ssid NOT DEFINED,
RSN=00F30006

Explanation: The DB2 subsystem is not active or not defined. Processing terminates.

User response: Verify that the DB2 subsystem is active and the correct DB2 subsystem ID is being used. If unable to resolve the problem, contact IBM Software Support.

CKZ23523I CONNECT TO DB2 SUBSYSTEM: ssid
VERSION: nnn

Explanation: A connection has been established to the DB2 subsystem. Processing continues.

User response: No action is required.

CKZ23524I NO APPLICATION DATA OR INDEX
SPACES NEED TO BE STARTED | NO
DB2 DIRECTORY OR CATALOG DATA
OR INDEX SPACES NEED TO BE
STARTED

Explanation: No application data or index spaces have an LPL or GRECP status or no DB2 directory or catalog data or index spaces have an LPL or GRECP status. Processing continues.

User response: No action is required.

CKZ23525E THERE ARE DB2 DIRECTORY OR
CATALOG DATA OR INDEX SPACES
THAT NEED TO BE STARTED

Explanation: The request was for application databases but there are DB2 directory or catalog data or index spaces that have an LPL or GRECP status. Processing terminates.

User response: Run DB2FIX with DATABASES(DB2) before running it with DATABASES(APPLICATION).

CKZ23526W DSNDB01.spname IS IN RESTRICTED
STATUS; DB2UPDATE NEEDS TO
RUN AGAIN

Explanation: The indicated space, DBD01 or SYSDBDXA, in database DSNDB01 had LPL or GRECP status. The starting of this space to fix the LPL or GRECP status has possibly caused the regression of updates made to it by DB2UPDATE. Processing continues.

User response: The DB2 subsystem needs to be stopped and DB2UPDATE run again with the DBD01ONLY keyword to redo the updates to DBD01 or SYSDBDXA.

CKZ23527E MEMBER: member HOLDS LOCKS; IT MUST BE STARTED TO RELEASE THE LOCKS

Explanation: The identified data sharing member was active during the cloning and holds locks against data or index spaces. Processing terminates.

User response: Start the identified data sharing member to release the locks. Then rerun DB2FIX.

CKZ23528E ERRORS PREVENT STARTING DATASPACE; MANUAL STARTS ARE NECESSARY

Explanation: DB2FIX is unable to start dataspace due to an error or condition that is beyond the capability of DB2FIX. Processing terminates.

User response: Manually identify and start the dataspace. Then contact IBM Software Support. Have available the listing that contains this message.

CKZ23529W MEMBER: member HOLDS LOCKS; DB2FIX MAY NEED TO BE RUN ON IT

Explanation: The identified data sharing member was active during the cloning and holds locks against table or index spaces. If the identified member has not been started, it needs to be started. If the identified member has been started, it may be necessary to also run DB2FIX on the identified member to resolve LPL or GRECP status. Processing continues.

User response: If DB2FIX fails to resolve LPL or GRECP status, run DB2FIX on the identified data sharing member using ACTION(CONTINUE) in the MEMBERS-NEED-STARTING keyword.

CKZ23530W NON NORMAL STATUS; DATABASE: database SPACENAM: spacename PART: partition number STATUS: status

Explanation: The identified space has a status that is restricted but does not include LPL or GRECP status. Processing continues.

User response: This space will require manual action to remove the restricted status.

CKZ23531E STATUS PREVENTS STARTING; DATABASE: database SPACENAM: spacename PART: partition number STATUS: status

Explanation: The identified space has a status that is restricted and has LPL or GRECP status and other restricted status. Processing terminates.

User response: This space will require manual action to remove the LPL or GRECP status.

CKZ23532W UNKNOWN RESPONSE LINE

Explanation: The response to a DB2 display command included an unknown line. Processing continues.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ23533E FORMAT OF RESPONSE LINE NOT AS EXPECTED

Explanation: The response to a DB2 display command included a line that did not have the expected format. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ23534E USERID IS NOT AUTHORIZED TO ISSUE THE DB2 COMMAND

Explanation: The userid running DB2FIX does not have the necessary authorization to issue the DB2 command. Processing terminates.

User response: The userid running DB2FIX needs to be defined as an install SYSADM in the DB2 zparms.

CKZ23535E COMMAND FAILED; RC= nnn

Explanation: The DB2 command failed to process. Nnn is the return code from the DB2 command processor. Processing terminates.

User response: If unable to determine the reason for the failure from the associated DB2 messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ23536E PAGE SPACE(S) DID NOT FIX WITHIN WAIT TIME LIMIT; DATABASE(dbname) SPACENAM(spname) clone

Explanation: The DB2 subsystem did not complete the start command of the indicated page space(s) to resolve the LPL or GRECP status within the time limit specified in the WAIT keyword. Processing terminates.

User response: Determine why the DB2 subsystem did not complete the start command. If the wait time limit is too short, increase it.

CKZ23537I WAITING FOR PAGE SPACE(S) TO FIX; DATABASE(dbname) SPACENAM(spname) clone

Explanation: DB2 Cloning Tool is waiting for the DB2 subsystem to complete the start command of the indicated page space(s) to resolve the LPL or GRECP status. Processing continues.

User response: No action is required.

**CKZ23538I PAGE SPACE(S) FIXED;
DATABASE(dbname)
SPACENAM(spname) clone**

Explanation: The DB2 subsystem has completed the start command of the indicated page space(s) and resolved the LPL or GRECP status. Processing continues.

User response: No action is required.

**CKZ23539E EXCESSIVE LOOPING; PROCESSING
TERMINATED**

Explanation: Excessive looping has been detected while trying to fix application page and index spaces. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ23540I DB2 COMMAND: db2 command

Explanation: Displays the command that will be executed by DB2. Processing continues.

User response: No action is required.

**CKZ23541I COMMAND NOT EXECUTED DUE TO
SIMULATION MODE**

Explanation: The DB2 command displayed in message CKZ23540I was not executed because this is a simulate run. Processing continues.

User response: No action is required.

**CKZ23542I dbname.spname WILL NOT BE
STARTED DUE TO EXCLUDE-MASKS
ENTRY: exclude-mask**

Explanation: The page space dbname.spname was found to have LPL or GRECP status but will not be started because there is an entry in the EXCLUDE-MASKS keyword it matches. Processing continues.

User response: The page space is not started.

**CKZ23547E CKZ00900 UNEXPECTED RESULTS;
error text**

Explanation: An unexpected condition occurred calling program CKZ00900. 'error text' has a description of the problem. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ23548I OPTIONS IN EFFECT FOR THIS
EXECUTION: merged options**

Explanation: Informational message indicating how DB2FIX will handle the options. The displayed options are derived from the CKZINI and any overriding specifications in the command input.

User response: No action is required.

**CKZ23550E ERROR IN PARAMETERS FOR
keyword**

Explanation: The parameters for the indicated keyword were incorrect. Processing terminates.

User response: Check the keyword parameters. Mutually exclusive keywords may have been used.

**CKZ23551E REQUIRED KEYWORD MISSING:
keyword**

Explanation: A keyword required for processing has been omitted. Processing terminates.

User response: Specify the required keyword.

**CKZ23553E KEYWORD: keyword MAXIMUM
LENGTH: nnn EXCEEDED**

Explanation: The operand entered for a keyword exceeded the maximum length allowed for the operand. nnn is the maximum allowed length for the keyword. Processing terminates.

User response: Correct the length of the keyword's operand.

**CKZ23554E KEYWORD HAS MORE THAN 1
OPERAND; ONLY 1 ALLOWED:
keyword**

Explanation: Multiple operands were detected for a keyword; only one operand is permitted. Processing terminates.

User response: Correct the keyword to use one operand.

**CKZ23556E NOTHING SPECIFIED FOR
KEYWORD: keyword**

Explanation: A keyword was entered without an appropriate operand. Processing terminates.

User response: Specify an appropriate operand for the keyword.

CKZ23558E INVALID VALUE IN KEYWORD:
keyword **VALUE: value error text**

Explanation: The value in the keyword is invalid. 'error text' indicates the problem detected with the value. Processing terminates.

User response: Correct the value specified in the keyword.

CKZ23585I VALUES FOR KEYWORD: keyword list of values

Explanation: Parsing found the listed values for the keyword.

User response: No action is required.

CKZ23586I VALIDATING KEYWORD: keyword

Explanation: Parsing is checking the indicated keyword indicated in the command.

User response: No action is required.

CKZ23590E INTERNAL ERROR; LOC=IIIII

Explanation: An internal processing error has occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ24001I hh:mm:ss DB2RBLDBSDS STARTED - PROGRAM REV=rrr | hh:mm:ss DB2RBLDBSDS COMPLETED; RETURN CODE=nnn

Explanation: DB2RBLDBSD processing message.

User response: No action is required.

CKZ24003I DDNAME=ddname ALLOCATED FOR DSN=datasetname

Explanation: 'ddname' has been dynamically allocated for the indicated data set.

User response: No action is required.

CKZ24004E DDNAME MISSING: ddname

Explanation: 'ddname' was specified for DB2 Cloning Tool to use. Processing terminates.

User response: Either correct the ddname specified, or add the appropriate ddname to the job's JCL.

CKZ24005E ALLOCATION FAILED FOR DSN: datasetname

Explanation: Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ24005W DEALLOCATION FAILED FOR DDNAME: ddname

Explanation: Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ24007W ERROR CALLING CKZ01HEX; FUNCTION: function R15=nnnn

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

CKZ24008E UNABLE TO LOAD PROGRAM: program name

Explanation: The indicated program name was not found. Processing terminates.

User response: Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

CKZ24009E ERROR ACCESSING JOURNAL FILE; LOC=IIIII

Explanation: A VSAM error occurred accessing a file. Processing terminates.

User response: See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ24011E JOURNAL CONTROL RECORD NOT FOUND | JOURNAL DB2 CONTROL RECORD NOT FOUND

Explanation: An expected record was not found in the DB2 Cloning Tool journal file. Processing terminates.

User response: Verify the same value was used for the DB2-NAME keyword that was used with the prior DB2UPDATE command for this DB2 subsystem or data sharing group. If unable to resolve the problem, Contact IBM Software Support. Have available the listing that contains this message.

CKZ24012E JOURNAL DB2 CONTROL RECORD IS WRONG VERSION

Explanation: The journal DB2 control record does not match the expected format. The record is printed. Processing terminates.

User response: Verify that different releases of DB2 Cloning Tool have not been run using the same journal data set. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ24020E NO DB2 GROUP NAME

Explanation: A DB2 group name was not found in the journal DB2 control record.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ24021E NO DB2 CATALOG NAME

Explanation: A DB2 catalog name was not found in the journal DB2 control record.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ24030I OPTIONS IN EFFECT FOR THIS EXECUTION: merged options

Explanation: Informational message indicating how DB2RBLDBSDS will handle the options. The displayed options are derived from the CKZINI and any overriding specifications in the command input.

User response: No action is required.

CKZ24051E REQUIRED KEYWORD MISSING: keyword

Explanation: A keyword required for processing has been omitted. Processing terminates.

User response: Specify the required keyword.

CKZ24053E KEYWORD: keyword MAXIMUM LENGTH: nnn EXCEEDED

Explanation: The operand entered for a keyword exceeded the maximum length allowed for the operand. nnn is the maximum allowed length for the keyword. Processing terminates.

User response: Correct the length of the keyword's operand.

CKZ24054E KEYWORD HAS MORE THAN 1 OPERAND; ONLY 1 ALLOWED: keyword

Explanation: Multiple operands were detected for a keyword; only one operand is permitted. Processing terminates.

User response: Correct the keyword to use one operand.

CKZ24056E NOTHING SPECIFIED FOR KEYWORD: keyword

Explanation: A keyword was entered without an appropriate operand. Processing terminates.

User response: Specify an appropriate operand for the keyword.

CKZ24058E INVALID VALUE IN KEYWORD: keyword VALUE: value error text

Explanation: The value in the keyword is invalid. 'error text' indicates the problem detected with the value. Processing terminates.

User response: Correct the value specified in the keyword.

CKZ24101I hh:mm:ss BSDS REBUILD STARTED - PROGRAM REV=rrr | hh:mm:ss BSDS REBUILD COMPLETED; RETURN CODE=nnn

Explanation: BSDS REBUILD processing message.

User response: No action is required.

CKZ24103I DDNAME=ddname ALLOCATED FOR DSN=datasetname

Explanation: 'ddname' has been dynamically allocated for the indicated data set.

User response: No action is required.

CKZ24104E OPEN FAILED FOR DDNAME=ddname RC=nnn VSAM RC=nnn

Explanation: Open has failed for ddname. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ24105E ALLOCATION FAILED FOR DSN: datasetname

Explanation: Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response: If unable to determine the reason for

the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ24105W DEALLOCATION FAILED FOR DDNAME: ddname

Explanation: Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ24107W ERROR CALLING CKZ01HEX; FUNCTION: function R15=nnnn

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

CKZ24109E ERROR ACCESSING JOURNAL FILE; LOC=IIIII

Explanation: A VSAM error occurred accessing a file. Processing terminates.

User response: See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ24111E JOURNAL DB2 MEMBER LOG RECORD NOT FOUND | JOURNAL DB2 MEMBER RECORD NOT FOUND

Explanation: An expected record was not found in the DB2 Cloning Tool journal file. Processing terminates.

User response: Verify the same value was used for the DB2-NAME keyword that was used with the prior DB2UPDATE command for this DB2 subsystem or data sharing group. If unable to resolve the problem, Contact IBM Software Support. Have available the listing that contains this message.

CKZ24112E JOURNAL DB2 MEMBER LOG RECORD IS WRONG VERSION | JOURNAL DB2 MEMBER RECORD IS WRONG VERSION | JOURNAL DB2 DDF RECORD IS WRONG VERSION

Explanation: The journal record does not match the expected format. The record is printed. Processing terminates.

User response: Verify that different releases of DB2 Cloning Tool have not been run using the same journal data set. If unable to resolve problem, contact IBM

Software Support. Have available the listing that contains this message.

CKZ24120E ERROR ACCESSING BSDS FILE; DDNAME=ddname LOC=location

Explanation: A VSAM error occurred accessing a BSDS file. Processing terminates.

User response: See the associated CKZVSEnnE error messages. If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ24121E BSDS CONTROL RECORD NOT FOUND

Explanation: The control record was not found in the BSDS. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ24122E BSDS CONTROL RECORD INCORRECT

Explanation: The control record that was found in the BSDS appears to be incorrect. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ24123I DSN=data_set_name1 COPIED TO DSN=data_set_name2

Explanation: The contents of the *data_set_name1* have been copied to *data_set_name2*.

User response: No action is required.

CKZ24130E DB2UPDATE HAS NOT BEEN RUN FOR THIS MEMBER

Explanation: DB2UPDATE has not been run for this member. processing terminates.

User response: Run DB2UPDATE for this member.

CKZ24131E DB2UPDATE FOR THIS MEMBER WAS SIMULATION, SIMULATION MUST BE SPECIFIED

Explanation: Simulation was not specified but the DB2UPDATE run for this member was a simulation.

User response: Run with simulation specified.

CKZ24135E DDNAME=ddname IS ALREADY ALLOCATED

Explanation: The ddname is already allocated and it must not be.

User response: Ensure the ddname is not allocated in

the JCL. If it is not in the JCL, contact IBM Software Support. Have available the listing that contains this message.

CKZ24136I DSN=datasetname HAS BEEN CLEARED

Explanation: All data has been removed from the data set.

User response: No action is required.

| **CKZ24140E UTILITY EXECUTION FAILED**

| **Explanation:** The invoked utility has failed.

| **User response:** : If unable to determine the reason for the failure from the associated messages, contact IBM Software Support. Have available the listing that contains this message.

CKZ24501I hh:mm:ss DB2SQL STARTED - PROGRAM REV=rrr | hh:mm:ss DB2SQL COMPLETED; RETURN CODE=nnn

Explanation: DB2SQL processing message.

User response: No action is required.

CKZ24503I DDNAME=ddname ALLOCATED FOR DSN=datasetname

Explanation: 'ddname' has been dynamically allocated for the indicated data set.

User response: No action is required.

CKZ24504E DDNAME MISSING: ddname

Explanation: 'ddname' was specified for DB2 Cloning Tool to use. Processing terminates.

User response: Either correct the ddname specified, or add the appropriate ddname to the job's JCL.

CKZ24505E ALLOCATION FAILED FOR DSN: datasetname

Explanation: Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ24505W DEALLOCATION FAILED FOR DDNAME: ddname

Explanation: Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ24507W ERROR CALLING CKZ01HEX; FUNCTION: function R15=nnnn

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

CKZ24508E UNABLE TO LOAD PROGRAM: program name

Explanation: The indicated program name was not found. Processing terminates.

User response: Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

CKZ24509E ERROR ACCESSING JOURNAL FILE; LOC=lllll

Explanation: A VSAM error occurred accessing a file. Processing terminates.

User response: See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

| **CKZ24511E JOURNAL record_type NOT FOUND**

| **Explanation:** An expected record was not found in the DB2 Cloning Tool journal file. The *record_type* listed in the message is the type of record that is not found. Processing terminates.

| **User response:** Verify the same value was used for the DB2-NAME keyword that was used with the prior DB2UPDATE command for this DB2 subsystem or data sharing group. If unable to resolve the problem, Contact IBM Software Support. Have available the listing that contains this message.

| **CKZ24512E JOURNAL record_type IS WRONG VERSION**

| **Explanation:** The journal record does not match the expected format. The incorrect *record_type* is listed in the message. The record is printed. Processing terminates.

| **User response:** Verify that different releases of DB2 Cloning Tool have not been run using the same journal data set. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains this message.

**CKZ24513E COUNT MISMATCH; RECORDS
READ=rrrr CONTROL RECORD
COUNT=cccc LOC=lllll**

Explanation: There was a problem with the journal records needed to initiate the command. The number of records read from the journal, rrrr, is not the same as the number indicated in the journal DB2 control record, cccc. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ24520E DSNALI ERROR; FUNCTION=
function RC=nnnn RSN=nnnnnnnn**

Explanation: An error occurred calling DSNALI. Function is the CAF function requested and RSN is the DB2 error reason. Processing terminates.

User response: Check the DB2 Messages and Codes manual for the RSN value. If unable to resolve the problem, contact IBM Software Support.

**CKZ24522E DB2 SUBSYSTEM: db2ssid NOT
ACTIVE, RSN=00F30002 | DB2
SUBSYSTEM: db2ssid NOT DEFINED,
RSN=00F30006**

Explanation: The DB2 subsystem is not active or not defined. Processing terminates.

User response: Verify that the DB2 subsystem is active and the correct DB2 subsystem ID is being used. If unable to resolve the problem, contact IBM Software Support.

**CKZ24523I CONNECT TO DB2 SUBSYSTEM: ssid
VERSION: nnn**

Explanation: A connection has been established to the DB2 subsystem. Processing continues.

User response: No action is required.

**CKZ24524E PLAN: planname NOT USABLE OR
MAY NOT EXIST, RSN=00F30040**

Explanation: A bind for this plan has not been done or the plan is not usable. Processing terminates.

User response: Bind the DB2 Cloning Tool plan to the DB2 subsystem.

**CKZ24548I OPTIONS IN EFFECT FOR THIS
EXECUTION: merged options**

Explanation: Informational message indicating how DB2SQL will handle the options. The displayed options are derived from the CKZINI and any overriding specifications in the command input.

User response: No action is required.

**CKZ24540E CKZ00900 UNEXPECTED RESULTS;
error text**

Explanation: An unexpected condition occurred calling program CKZ00900. 'error text' has a description of the problem. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ24550E ERROR IN PARAMETERS FOR
keyword**

Explanation: The parameters for the indicated keyword were incorrect. Processing terminates.

User response: Check the keyword parameters. Mutually exclusive keywords may have been used.

**CKZ24551E REQUIRED KEYWORD MISSING:
keyword**

Explanation: A keyword required for processing has been omitted. Processing terminates.

User response: Specify the required keyword.

**CKZ24552E REQUIRED INI SECTION/TOKEN
MISSING: SECTION=section
TOKEN=token | REQUIRED INI
VALUE MISSING FOR
SECTION=section TOKEN=token |
INVALID INI VALUE FOR
SECTION=section TOKEN=token**

Explanation: An error occurred validating the CKZINI member options. Processing terminates.

User response: Correct the CKZINI member.

**CKZ24553E KEYWORD: keyword MAXIMUM
LENGTH: nnn EXCEEDED**

Explanation: The operand entered for a keyword exceeded the maximum length allowed for the operand. nnn is the maximum allowed length for the keyword. Processing terminates.

User response: Correct the length of the keyword's operand.

CKZ24554E KEYWORD HAS MORE THAN 1 OPERAND; ONLY 1 ALLOWED: keyword

Explanation: Multiple operands were detected for a keyword; only one operand is permitted. Processing terminates.

User response: Correct the keyword to use one operand.

CKZ24556E NOTHING SPECIFIED FOR KEYWORD: keyword

Explanation: A keyword was entered without an appropriate operand. Processing terminates.

User response: Specify an appropriate operand for the keyword.

CKZ24558E INVALID VALUE IN KEYWORD: keyword VALUE: value error text

Explanation: The value in the keyword is invalid. 'error text' indicates the problem detected with the value. Processing terminates.

User response: Correct the value specified in the keyword.

CKZ24575E UNMATCHED ENTRIES IN KEYWORD: keyword

Explanation: For WLM-ENVIRONMENT-MASKS there must be pairs of source and target masks. An uneven number of entries was specified. Processing terminates.

User response: Correct the keyword specification.

CKZ24576E THE USE OF keyword REQUIRES DB2 V9

Explanation: The keyword can only be used with DB2 Version 9.1 and the DB2 system is less than DB2 Version 9.1. Processing terminates.

User response: Remove the keyword.

CKZ24585I PAIRS FOR KEYWORD: keyword list of pairs

Explanation: Parsing found the listed pairs for the keyword.

User response: No action is required.

CKZ24586I VALIDATING KEYWORD: keyword

Explanation: Parsing is checking the indicated keyword indicated in the command.

User response: No action is required.

CKZ24601I hh:mm:ss SQL PROCESSOR STARTED - PROGRAM REV=rrr | hh:mm:ss SQL PROCESSOR COMPLETED; RETURN CODE=nnn

Explanation: SQL PROCESSOR processing message.

User response: No action is required.

CKZ24606E ERROR CALLING CKZ01VV1 tttttt FUNCTION: function R15=nnnn R0=nnnnnnnn LOC=lllll

Explanation: A problem occurred using a dataspace. tttttt is the name of the internal table. lllll is the location where the error occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message and the parmlib member that controls execution of DB2 Cloning Tool.

CKZ24607W ERROR CALLING CKZ01HEX; FUNCTION: function R15=nnnn

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

CKZ24620W DSNTIAR FAILED; RC:nnnn RSN: nnnnnnnn

Explanation: An error occurred using DSNTIAR to format error messages. Processing continues.

User response: Please report this message to IBM Software Support.

CKZ24630I type INFORMATION UPDATED IN TABLE: tablename CHANGED ROWS: nnnnnn | type INFORMATION WOULD HAVE BEEN UPDATED IN TABLE: tablename CHANGED ROWS: nnnnnn

Explanation: The table has been updated, or would have been updated, with the indicated information type. Processing continues.

User response: No action is required.

CKZ24631I text

Explanation: Displays the generated SQL statement being processed. Processing continues.

User response: No action is required.

CKZ24632I *New | Old DB2 STOGROUPS created |
dropped: number_created_or_dropped | New
| Old DB2 STOGROUPS would have
been created | dropped:
number_created_or_dropped*

Explanation: The new name DB2 STOGROUPS have been or would have been created, or the old name DB2 STOGROUPS have been or would have been dropped. Processing continues.

User response: No action is required.

CKZ24633I *New | Old DB2 STOGROUP already
created | dropped: stogroup-name .*

Explanation: New name DB2 STOGROUPS have already been created, or old name DB2 STOGROUPS have already been dropped. Processing continues.

User response: No action is required.

CKZ24634I **Volume *volser* in STOGROUP
stogroup-name is not a source volume
New STOGROUP *new-stogroup-name***

Explanation: A *volser* in DB2 STOGROUP *stogroup-name* is not a source volume of the cloning. The new DB2 STOGROUP *new-stogroup-name* being created will have this volume. Processing continues.

User response: No action is required. You might want to verify that the *volser* is intended to be used in the DB2 STOGROUP. Manually remove the volume from the DB2 STOGROUP if desired.

CKZ24650W **item value NOT CHANGED, IT DOES
NOT MATCH KEYWORD**

Explanation: For item of WLM ENVIRONMENT, the WLM_ENVIRONMENT value found in SYSIBM.SYSROUTINES did not match a source mask specified in the WLM-ENVIRONMENT-MASKS keyword. For item of DATACLAS, the DATACLAS value found in SYSIBM.SYSSTOGROUP did not match a source mask specified in the DATACLAS-MASKS keyword. For item of MGMTCLAS, the MGMTCLAS value found in SYSIBM.SYSSTOGROUP did not match a source mask specified in the MGMTCLAS-MASKS keyword. For item of STORCLAS, the STORCLAS value found in SYSIBM.SYSSTOGROUP did not match a source mask specified in the STORCLAS-MASKS keyword. Processing continues.

User response: Correct the WLM-ENVIRONMENT-MASKS, DATACLAS-MASKS, MGMTCLAS-MASKS, or STORCLAS-MASKS names if they are in error.

CKZ24651W **WLM ENVIRONMENT NOT
CHANGED FOR *procedure-name*
UNABLE TO ESTABLISH NEEDED
ENVIRONMENT SETTINGS FOR
ALTER**

Explanation: An attempt to change the WLM ENVIRONMENT name for a native SQL procedure failed. The ALTER procedure SQL statement being used requires that the environment settings in effect must be the same as when the procedure was first created. DB2 Cloning Tool changes the settings of CURRENT PATH and CURRENT SCHEMA to match what they were at the time the procedure was created. Other environment settings can come from system definitions such as the DSNHDECP module. If these other environment settings do not match what they were at the time the procedure was created the ALTER will fail with a SQLCODE of -4706. A DSNT408I message should be displayed that shows the ENVID of the environment that existed at the time of the ALTER and the ENVID of the environment that existed when the procedure was created. Processing continues.

User response: The WLM ENVIRONMENT name for this procedure will need to be changed manually. The ENVIDs in the DSNT408I message can be used for selecting the corresponding rows from SYSIBM.SYSENVIRONMENT. Comparing the two rows will show which environment settings are different and need to be set for an ALTER statement to work for this procedure.

CKZ24652W **ENVID=*nnn* NOT FOUND IN
SYSIBM.SYSENVIRONMENT; UNABLE
TO SET ENVIRONMENT FOR ALTER**

Explanation: The row in SYSIBM.SYSENVIRONMENT for the ENVID of a native SQL procedure was not found. This condition could indicate a problem with the DB2 catalog and tables SYSIBM.SYSENVIRONMENT and SYSIBM.SYSROUTINES. There should be a following CKZ24651W message that will identify the procedure name. Processing continues.

User response: The WLM ENVIRONMENT name for this procedure will need to be changed manually. The reason for the missing row in SYSIBM.SYSROUTINES should be determined and corrected.

CKZ24653I **ALTER RECEIVED SQLCODE: -449;
WILL ADD EXTERNAL NAME AND
RETRY ALTER**

Explanation: The ALTER statement received a SQLCODE -449. This SQLCODE indicates that EXTERNAL NAME must be included in the ALTER statement. EXTERNAL NAME will be added to the ALTER statement and the modified ALTER will be executed.

| **User response:** No action is required.

CKZ25001I hh:mm:ss DB2LGRNXCLEAN
 STARTED - PROGRAM REV=rrr |
 hh:mm:ss DB2LGRNXCLEAN
 COMPLETED; RETURN CODE=nnn

Explanation: DB2LGRNXCLEAN processing message.

User response: No action is required.

CKZ25003I DDNAME=ddname ALLOCATED FOR
 DSN=datasetname

Explanation: 'ddname' has been dynamically allocated for the indicated data set.

User response: No action is required.

CKZ25004E DDNAME MISSING: ddname

Explanation: 'ddname' was specified for DB2 Cloning Tool to use. Processing terminates.

User response: Either correct the ddname specified, or, add the appropriate ddname to the job's JCL.

CKZ25005E ALLOCATION FAILED FOR DSN:
 datasetname

Explanation: Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ25005W DEALLOCATION FAILED FOR
 DDNAME: ddname

Explanation: Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

| **CKZ25007W** ERROR CALLING CKZ01HEX;
 | FUNCTION=function R15=nnnn

| **Explanation:** An error occurred using CKZ01HEX to print a record. Processing continues.

| **User response:** Report this message to IBM Software Support.

CKZ25008E UNABLE TO LOAD PROGRAM:
 program name

Explanation: The indicated program name was not found. Processing terminates.

User response: Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

CKZ25009E ERROR ACCESSING JOURNAL FILE;
 LOC=IIIII

Explanation: A VSAM error occurred accessing a file. Processing terminates.

User response: See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ25011E JOURNAL *type* RECORD NOT FOUND

Explanation: An expected record was not found in the DB2 Cloning Tool journal file. Processing terminates.

User response: Verify the same value was used for the DB2-NAME keyword that was used with the prior DB2UPDATE command for this DB2 subsystem or data sharing group. If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ25012E JOURNAL *type* RECORD IS WRONG
 VERSION

Explanation: A journal record does not match the expected format. The record is printed. Processing terminates.

User response: Verify that different releases of DB2 Cloning Tool have not been run using the same journal data set. A journal created by a prior release of DB2 Cloning Tool can be upgraded to the current release by using the JRNLUPGRADE command. See the "JRNLUPGRADE" on page 428 topic for the scenarios where JRNLUPGRADE can be used. If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ25020E NO DB2 CATALOG NAME

Explanation: A DB2 catalog name was not found in the journal DB2 control record.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ25030I OPTIONS IN EFFECT FOR THIS EXECUTION: merged options

Explanation: Informational message indicating how DB2RBLDBSDS will handle the options. The displayed options are derived from the CKZINI and any overriding specifications in the command input.

User response: No action is required.

CKZ25031E DB2UPDATES WERE ALL SIMULATIONS, SIMULATION MUST BE SPECIFIED

Explanation: All DB2UPDATE runs were simulations so this must be a simulation. Processing terminates.

User response: Run with simulation specified.

CKZ25051E REQUIRED KEYWORD MISSING: keyword

Explanation: A keyword required for processing has been omitted. Processing terminates.

User response: Specify the required keyword.

CKZ25053E KEYWORD: keyword MAXIMUM LENGTH: nnn EXCEEDED

Explanation: The operand entered for a keyword exceeded the maximum length allowed for the operand. nnn is the maximum allowed length for the keyword. Processing terminates.

User response: Correct the length of the keyword's operand.

CKZ25054E KEYWORD HAS MORE THAN 1 OPERAND; ONLY 1 ALLOWED: keyword

Explanation: Multiple operands were detected for a keyword; only one operand is permitted. Processing terminates.

User response: Correct the keyword to use one operand.

CKZ25056E NOTHING SPECIFIED FOR KEYWORD: keyword

Explanation: A keyword was entered without an appropriate operand. Processing terminates.

User response: Specify an appropriate operand for the keyword.

CKZ25058E INVALID VALUE IN KEYWORD: keyword VALUE: value error text

Explanation: The value in the keyword is invalid. 'error text' indicates the problem detected with the value. Processing terminates.

User response: Correct the value specified in the keyword.

CKZ25101I hh:mm:ss SYSLGRNX CLEAN STARTED - PROGRAM REV=rrr | hh:mm:ss SYSLGRNX CLEAN COMPLETED; RETURN CODE=nnn

Explanation: SYSLGRNX CLEAN processing message.

User response: No action is required.

CKZ25103I DDNAME=ddname ALLOCATED FOR DSN=datasetname

Explanation: 'ddname' has been dynamically allocated for the indicated data set.

User response: No action is required.

CKZ25104E OPEN FAILED FOR DDNAME=ddname

Explanation: Open has failed for ddname. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ25105E ALLOCATION FAILED FOR DSN: datasetname

Explanation: Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ25105W DEALLOCATION FAILED FOR DDNAME: ddname

Explanation: Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ25107W ERROR CALLING CKZ01HEX;
FUNCTION=*function* R15=*nnnn*

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Report this message to IBM Software Support.

CKZ25108E UNABLE TO LOAD PROGRAM:
program name

Explanation: The indicated program name was not found. Processing terminates.

User response: Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

CKZ25121E DDNAME=*ddname* IS ALREADY ALLOCATED

Explanation: The *ddname* is already allocated and it must not be.

User response: Ensure the *ddname* is not allocated in the JCL. If it is not in the JCL, contact IBM Software Support. Have available the listing that contains this message.

CKZ25122I SPACE=*spacename* DSN=*datasetname* HAS BEEN CLEANED

Explanation: All data has been removed from the data set.

User response: No action is required.

CKZ25124I SPACE=*spacename* HAS *attribute*

Explanation: The identified table or index space has the indicated attribute.

User response: No response is required.

CKZ25132I DATA FROM DDNAME=*ddname* IS BEING USED

Explanation: Data is being used from the file allocated to *ddname*.

User response: No action is required.

CKZ25131E DSNUGICR UTILITY EXECUTION FAILED

Explanation: The invoked DSNUGICR utility has failed.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ25141E UNABLE TO DETERMINE IF EXTENDED RBA IS ENABLED FOR DSNDB01.*tsname*.

Explanation: : It cannot be determined whether the indicated table or index space that is in the DB2 directory was enabled for extended RBA. It is not possible to properly clean the table or index space. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ25201I *hh:mm:ss* DB2UTILXCLEAR STARTED - PROGRAM REV=*rrr* | *hh:mm:ss* DB2UTILXCLEAR COMPLETED; RETURN CODE=*nnn*

Explanation: This is an informational DB2UTILXCLEAR processing message.

User response: No response is required.

CKZ25203I DDNAME=*ddname* ALLOCATED FOR DSN=*data_set_name*

Explanation: The *ddname* that is listed in the message text was dynamically allocated for the indicated data set.

User response: No response is required.

CKZ25204E DDNAME MISSING: *ddname*

Explanation: The *ddname* that is listed in the message text was specified for DB2 Cloning Tool to use, but is missing. Processing terminates.

User response: Either correct the specified *ddname*, or add the appropriate *ddname* to the job's JCL.

CKZ25205E ALLOCATION FAILED FOR DSN: *data_set_name*

Explanation: Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing that contains these messages.

CKZ25205W DEALLOCATION FAILED FOR DDNAME: *ddname*

Explanation: Dynamic deallocation for the *ddname* that is listed in the message failed. The associated z/OS messages are displayed. Processing continues.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact

| IBM Software Support. Have available the listing that
| contains these messages.

**CKZ25207W ERROR CALLING CKZ01HEX;
FUNCTION: *function* R15=*nnnn***

| **Explanation:** An error occurred while CKZ01HEX was
| attempting to print a record. Processing continues.

| **User response:** If unable to determine the reason for
| the failure, contact IBM Software Support.

**CKZ25208E UNABLE TO LOAD PROGRAM:
*program_name***

| **Explanation:** The program name that is listed in the
| message text was not found. Processing terminates.

| **User response:** Check that the job's //STEPLIB library
| is correct. If unable to resolve the problem, contact IBM
| Software Support.

**CKZ25209E ERROR ACCESSING JOURNAL FILE;
LOC=*location***

| **Explanation:** A VSAM error occurred accessing a file.
| Processing terminates.

| **User response:** Refer to the associated CKZVSEnnE
| error messages. If unable to resolve the problem,
| contact IBM Software Support. Have available the
| listing that contains these messages.

CKZ25211E JOURNAL *type* RECORD NOT FOUND

| **Explanation:** An expected record was not found in the
| DB2 Cloning Tool journal file. Processing terminates.

| **User response:** Verify that the same value was used
| for the DB2-NAME keyword that was used with the
| prior DB2UPDATE command for this DB2 subsystem
| or data sharing group. If unable to resolve the problem,
| contact IBM Software Support. Have available the
| listing that contains these messages.

**CKZ25212E JOURNAL *type* RECORD IS WRONG
VERSION**

| **Explanation:** The identified journal record does not
| match the expected format. The record is printed.
| Processing terminates.

| **User response:** Verify that different releases of DB2
| Cloning Tool were not run using the same journal data
| set. If unable to resolve the problem, contact IBM
| Software Support. Have available the listing that
| contains these messages.

CKZ25220E NO DB2 CATALOG NAME

| **Explanation:** A DB2 catalog name was not found in
| the journal DB2 control record.

| **User response:** Contact IBM Software Support. Have
| available the listing that contains these messages.

**CKZ25230I OPTIONS IN EFFECT FOR THIS
EXECUTION: *merged options***

| **Explanation:** This informational message indicates
| how DB2UTILXCLEAN will handle the options. The
| displayed options are derived from the INI file and any
| overriding specifications in the command input.

| **User response:** No action is required.

**CKZ25231E DB2UPDATES WERE ALL
SIMULATIONS, SIMULATION MUST
BE SPECIFIED**

| **Explanation:** All DB2UPDATE runs were simulations,
| so this run must be a simulation. Processing terminates.

| **User response:** Run with simulation (SIM) specified.

**CKZ25251E REQUIRED KEYWORD MISSING:
*keyword***

| **Explanation:** A keyword that is required for
| processing was omitted. Processing terminates.

| **User response:** Specify the required keyword.

**CKZ25253E KEYWORD: *keyword* MAXIMUM
LENGTH: *length* EXCEEDED**

| **Explanation:** The operand entered for a keyword
| exceeded the maximum length that is allowed for the
| operand. *length* is the maximum allowed length for the
| keyword. Processing terminates.

| **User response:** Correct the length of the keyword's
| operand.

**CKZ25254E KEYWORD HAS MORE THAN 1
OPERAND; ONLY 1 ALLOWED:
*keyword***

| **Explanation:** Multiple operands were detected for a
| keyword; only one operand is allowed. Processing
| terminates.

| **User response:** Correct the keyword to use one
| operand.

**CKZ25256E NOTHING SPECIFIED FOR
KEYWORD: *keyword***

| **Explanation:** A keyword was entered without an
| appropriate operand. Processing terminates.

| **User response:** Specify an appropriate operand for the keyword.

| **CKZ25258E INVALID VALUE IN KEYWORD:**
| *keyword* **VALUE:** *value error_text*

| **Explanation:** The value in the keyword that is listed in the message is invalid. *error_text* indicates the problem that was detected with the value. Processing terminates.

| **User response:** Correct the value that is specified in the keyword.

| **CKZ25301I** *hh:mm:ss* **SYSUTILX CLEAN STARTED -**
| **PROGRAM REV=rrr** | *hh:mm:ss*
| **SYSUTILX CLEAN COMPLETED;**
| **RETURN CODE=return_code**

| **Explanation:** This message is a SYSUTILX CLEAN processing informational message.

| **User response:** No action is required.

| **CKZ25303I DDNAME=ddname ALLOCATED FOR**
| **DSN=data_set_name**

| **Explanation:** *ddname* was dynamically allocated for the indicated data set.

| **User response:** No action is required.

| **CKZ25304E OPEN FAILED FOR DDNAME=ddname**

| **Explanation:** Open failed for the *ddname* that is listed in the message. Processing terminates.

| **User response:** Contact IBM Software Support. Have available the listing that contains this message.

| **CKZ25305E ALLOCATION FAILED FOR DSN:**
| *data_set_name*

| **Explanation:** Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

| **User response:** If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing that contains these messages.

| **CKZ25305W DEALLOCATION FAILED FOR**
| **DDNAME: ddname**

| **Explanation:** Dynamic deallocation for a *ddname* failed. The associated z/OS messages are displayed. Processing continues.

| **User response:** If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing that contains these messages.

| **CKZ25307W ERROR CALLING CKZ01HEX;**
| **FUNCTION:** *function* **R15=nnnn**

| **Explanation:** An error occurred while CKZ01HEX was attempting to print a record. Processing continues.

| **User response:** If unable to determine the reason for the failure, contact IBM Software Support.

| **CKZ25308E UNABLE TO LOAD PROGRAM:**
| *program_name*

| **Explanation:** The indicated program name was not found. Processing terminates.

| **User response:** Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

| **CKZ25321E DDNAME=ddname IS ALREADY**
| **ALLOCATED**

| **Explanation:** The *ddname* is already allocated and it must not be.

| **User response:** Ensure that the *ddname* is not allocated in the JCL. If it is not in the JCL, contact IBM Software Support. Have available the listing that contains this message.

| **CKZ25322I SPACE=spacename DSN=data_set_name**
| **HAS BEEN CLEANED**

| **Explanation:** All data was removed from the data set.

| **User response:** No action is required.

| **CKZ25323I DATA FROM DDNAME=ddname IS**
| **BEING USED**

| **Explanation:** Data is being used from the file allocated to *ddname*.

| **User response:** No action is required.

| **CKZ25324I SPACE=spacename HAS attribute**

| **Explanation:** The identified table or index space has the indicated attribute.

| **User response:** No action is required.

| **CKZ25331E DSNUGICR UTILITY EXECUTION**
| **FAILED**

| **Explanation:** The invoked DSNUGICR utility failed.

| **User response:** Contact IBM Software Support. Have available the listing that contains this message.

CKZ25341E UNABLE TO DETERMINE IF EXTENDED RBA IS ENABLED FOR DSNDB01.tsnam

Explanation: It cannot be determined whether the indicated table or index space in the DB2 directory has been enabled for extended RBA. It is not possible to properly clean the table or index space. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ25401I hh:mm:ss EXTENDED RBA CHECKER STARTED - PROGRAM REV=rrr | hh:mm:ss EXTENDED RBA CHECKER COMPLETED; RETURN CODE=return_code

Explanation: Informational EXTENDED RBA CHECKER processing message.

User response: No action is required.

CKZ25403I DDNAME=ddname ALLOCATED FOR DSN=data_set_name

Explanation: The ddname that is listed in the message was dynamically allocated for the indicated data set.

User response: No action is required.

CKZ25405E ALLOCATION FAILED FOR DSN: data_set_name

Explanation: Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing that contains these messages.

CKZ25405W DEALLOCATION FAILED FOR DDNAME: ddname

Explanation: Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing that contains these messages.

CKZ25407W ERROR CALLING CKZ01HEX; FUNCTION: function R15=nnnn

Explanation: An error occurred while CKZ01HEX was attempting to print a record. Processing continues.

User response: If unable to determine the reason for

the failure, contact IBM Software Support.

CKZ25410E ERROR ACCESSING LINEAR FILE; DDNAME=ddname R15=nnnn ERROR=nnn FUNCTION=function LOC=location RBA OF RECORD: X'nnnnnnnn_nnnnnnnn'

Explanation: A VSAM error occurred accessing the indicated file. Processing terminates.

User response: If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ25442I DSNDB01.tsnam DDNAME=ddname IS EMPTY

Explanation: The indicated table or index space in the DB2 directory is empty.

User response: No action is required.

CKZ25501I hh:mm:ss DB2START STARTED - PROGRAM REV=rrr | hh:mm:ss DB2START COMPLETED; RETURN CODE=nnn

Explanation: DB2START processing message.

User response: No action is required.

CKZ25507W ERROR CALLING CKZ01HEX; FUNCTION: function R15=nnnn

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

CKZ25508E UNABLE TO LOAD PROGRAM: program name

Explanation: The indicated program name was not found. Processing terminates.

User response: Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

CKZ25520E DSNALI ERROR; FUNCTION=function RC=nnnn RSN=nnnnnnnn

Explanation: An error occurred calling DSNALI. Function is the CAF function requested and RSN is the DB2 error reason. Processing terminates.

User response: Check the DB2 Messages and Codes manual for the RSN value. If unable to resolve the problem, contact IBM Software Support.

CKZ25522E DB2 SUBSYSTEM: db2ssid NOT DEFINED | DB2 SUBSYSTEM: db2ssid NOT DEFINED, RSN=00F30006

Explanation: The DB2 subsystem is not defined to z/OS. Processing terminates.

User response: Verify that the correct DB2 subsystem ID is being used. If unable to resolve the problem, contact IBM Software Support.

CKZ25523I CONNECTED TO DB2 SUBSYSTEM: ssid VERSION: nnn

Explanation: A connection has been established to the DB2 subsystem. Processing continues.

User response: No action is required.

CKZ25535E DB2 STOPPED PREMATURELY

Explanation: The DB2 subsystem stopped prematurely. The DB2 subsystem stopped before it became ready. Processing terminates.

User response: Determine why the DB2 subsystem did not complete its start up.

CKZ25536E name DID NOT START WITHIN keyword TIME LIMIT

Explanation:

The DB2 subsystem did not complete its start up within the time limit specified in the MSTR-DETECT-WAIT or WAIT keyword. For the MSTR-DETECT-WAIT keyword, the xxxMSTR address space did not start executing on z/OS. The xxxMSTR address space may have experienced a JCL error. For the WAIT keyword, the xxxMSTR address space is executing on z/OS, but DB2 did not become ready for work. Processing terminates.

User response: Determine why the DB2 subsystem did not complete its start up. If the wait time limit is too short, increase it.

CKZ25537I DB2 HAS STARTED | DB2 HAS TERMINATED

Explanation: The DB2 subsystem has started or terminated. Processing continues.

User response: No action is required.

CKZ25538I WAITING FOR DB2 TO START | WAITING FOR DB2 TO TERMINATE

Explanation: DB2 Cloning Tool is waiting for the DB2 subsystem to start or terminate. Processing continues.

User response: No action is required.

CKZ25539E DB2 IS ALREADY RUNNING

Explanation: The DB2 subsystem is already running. Processing terminates.

User response: Determine why the DB2 subsystem is running.

CKZ25540I START COMMAND: command

Explanation: Displays the start command that will be issued. Processing continues.

User response: No action is required.

CKZ25541I COMMAND NOT EXECUTED DUE TO SIMULATION MODE

Explanation: The start command was not issued because this is a simulation run. Processing continues.

User response: No action is required.

CKZ25542E UNABLE TO DETERMINE DB2 STATUS; DB2 IS RUNNING BUT IS NOT ACCEPTING CONNECTIONS

Explanation: DB2 Cloning Tool is unable to determine the status of the DB2 subsystem. The DB2 subsystem is running but is not accepting connections. Processing terminates.

User response: Get the DB2 subsystem either all the way up or stop it.

CKZ25543I RESTART REPLY: reply

Explanation: Displays the restart reply that will be issued.

User response: No action is required.

CKZ25544I REPLY NOT EXECUTED DUE TO SIMULATION MODE

Explanation: The restart reply was not issued because this is a simulation run. Processing continues.

User response: No action is required.

CKZ25545E macro ERROR; FUNCTION=function RC=nnnn RSN=X'nnnnnnnn'

Explanation: An error occurred calling a system macro. Function is the macro function requested, RC is the return code, and RSN is the reason. Processing terminates.

User response: If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ25546E INTERNAL ERROR; LOC=lllll reason text

Explanation: An internal error has occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ25548I OPTIONS IN EFFECT FOR THIS EXECUTION: merged options

Explanation: Informational message indicating how DB2START will handle the options. The displayed options are derived from the CKZINI and any overriding specifications in the command input.

User response: No action is required.

CKZ25550E ERROR IN PARAMETERS FOR keyword

Explanation: The parameters for the indicated keyword were incorrect. Processing terminates.

User response: Check the keyword parameters. Mutually exclusive keywords may have been used.

CKZ25551E REQUIRED KEYWORD MISSING: keyword

Explanation: A keyword required for processing has been omitted. Processing terminates.

User response: Specify the required keyword.

CKZ25553E KEYWORD: keyword MAXIMUM LENGTH: nnn EXCEEDED

Explanation: The operand entered for a keyword exceeded the maximum length allowed for the operand. nnn is the maximum allowed length for the keyword. Processing terminates.

User response: Correct the length of the keyword's operand.

CKZ25554E KEYWORD HAS MORE THAN 1 OPERAND; ONLY 1 ALLOWED: keyword

Explanation: Multiple operands were detected for a keyword; only one operand is permitted. Processing terminates.

User response: Correct the keyword to use one operand.

CKZ25556E NOTHING SPECIFIED FOR KEYWORD: keyword

Explanation: A keyword was entered without an appropriate operand. Processing terminates.

User response: Specify an appropriate operand for the keyword.

CKZ25558E INVALID VALUE IN KEYWORD: keyword VALUE: value error text

Explanation: The value in the keyword is invalid. 'error text' indicates the problem detected with the value. Processing terminates.

User response: Correct the value specified in the keyword.

CKZ25560E section NOT AS EXPECTED

Explanation: The section specified does not have the expected format. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ25561E SPECIAL DSNZPARM DOES NOT HAVE DB2 CATALOG UPDATABLE ATTRIBUTE

Explanation: The special dsnzparm does not have the DB2 catalog updatable attribute. Processing terminates.

User response: Correct the special dsnzparm to have the DB2 catalog updatable attribute.

CKZ25562E SPECIAL DSNZPARM DOES NOT HAVE DEFER,ALL ATTRIBUTE

Explanation: The special dsnzparm does not have the DEFER,ALL attribute. Processing terminates.

User response: Correct the special dsnzparm to have the DEFER,ALL attribute.

CKZ25563E UNABLE TO LOAD SPECIAL DSNZPARM MODULE

Explanation: The specified dsnzparm module was not able to be loaded. Processing terminates.

User response: Verify the //STEPLIB includes the load library the contains the special dsnzparm.

CKZ25564E DSNZPARM MUST BE SPECIFIED WHEN SPECIAL IS SPECIFIED

Explanation: The SPECIAL keyword was specified but the DSNZPARM keyword was not specified. Processing terminates.

User response: Add the DSNZPARM keyword with the name of the special dsnzparm module.

**CKZ25565I WAIT TIME IS LESS THAN
MSTR-DETECT-WAIT TIME - WAIT
TIMEOUT WILL HAPPEN BEFORE
MSTR-DETECT-WAIT TIMEOUT**

Explanation: The time specified in the WAIT keyword is less than the time specified in the MSTR-DETECT-WAIT keyword. This will cause the MSTR-DETECT-WAIT timeout to be disabled, as the WAIT timeout will happen before the MSTR-DETECT-WAIT timeout can happen.

User response: No action is required. If desired, the MSTR-DETECT-WAIT time can be changed to be less than the WAIT time.

CKZ25570I STARTING DB2 SUBSYSTEM: ssid

Explanation: The DB2 subsystem is being started. Processing continues.

User response: No action is required.

CKZ25586I VALIDATING KEYWORD: keyword

Explanation: Parsing is checking the indicated keyword indicated in the command.

User response: No action is required.

**CKZ26001I hh:mm:ss DB2XCFCLEAN STARTED -
PROGRAM REV=rrr | hh:mm:ss
DB2XCFCLEAN COMPLETED;
RETURN CODE=nnn**

Explanation: DB2XCFCLEAN processing message.

User response: No action is required.

**CKZ26003I DDNAME=ddname ALLOCATED FOR
DSN=datasetname**

Explanation: 'ddname' has been dynamically allocated for the indicated data set.

User response: No action is required.

CKZ26004E DDNAME MISSING: ddname

Explanation: 'ddname' was specified for DB2 Cloning Tool to use. Processing terminates.

User response: Either correct the ddname specified, or add the appropriate ddname to the job's JCL.

**CKZ26005E ALLOCATION FAILED FOR DSN:
datasetname**

Explanation: Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact

IBM Software Support. Have available the listing containing these messages.

**CKZ26005W DEALLOCATION FAILED FOR
DDNAME: ddname**

Explanation: Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ26008E UNABLE TO LOAD PROGRAM:
program name**

Explanation: The indicated program name was not found. Processing terminates.

User response: Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

**CKZ26009E ERROR ACCESSING JOURNAL FILE;
LOC=IIIIII**

Explanation: A VSAM error occurred accessing a file. Processing terminates.

User response: See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ26011E JOURNAL DB2 CONTROL RECORD
NOT FOUND**

Explanation: An expected record was not found in the DB2 Cloning Tool journal file. Processing terminates.

User response: Verify the same value was used for the DB2-NAME keyword that was used with the prior DB2UPDATE command for this DB2 subsystem or data sharing group. If unable to resolve the problem, Contact IBM Software Support. Have available the listing that contains this message.

**CKZ26012E JOURNAL DB2 CONTROL RECORD IS
WRONG VERSION**

Explanation: The journal DB2 control record does not match the expected format. The record is printed. Processing terminates.

User response: Verify that different releases of DB2 Cloning Tool have not been run using the same journal data set. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ26020E NO DB2 GROUP NAME

Explanation: A DB2 group name was not found in the journal DB2 control record.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ26048I OPTIONS IN EFFECT FOR THIS EXECUTION: merged options

Explanation: Informational message indicating how DB2XCFCLEAN will handle the options. The displayed options are derived from the CKZINI and any overriding specifications in the command input.

User response: No action is required.

CKZ26051E REQUIRED KEYWORD MISSING: keyword

Explanation: A keyword required for processing has been omitted. Processing terminates.

User response: Specify the required keyword.

CKZ26053E KEYWORD: keyword MAXIMUM LENGTH: nnn EXCEEDED

Explanation: The operand entered for a keyword exceeded the maximum length allowed for the operand. nnn is the maximum allowed length for the keyword. Processing terminates.

User response: Correct the length of the keyword's operand.

CKZ26054E KEYWORD HAS MORE THAN 1 OPERAND; ONLY 1 ALLOWED: keyword

Explanation: Multiple operands were detected for a keyword; only one operand is permitted. Processing terminates.

User response: Correct the keyword to use one operand.

CKZ26056E NOTHING SPECIFIED FOR KEYWORD: keyword

Explanation: A keyword was entered without an appropriate operand. Processing terminates.

User response: Specify an appropriate operand for the keyword.

CKZ26058E INVALID VALUE IN KEYWORD: keyword VALUE: value error text

Explanation: The value in the keyword is invalid. 'error text' indicates the problem detected with the value. Processing terminates.

User response: Correct the value specified in the keyword.

CKZ26501I hh:mm:ss DB2STOP STARTED - PROGRAM REV=rrr | hh:mm:ss DB2STOP COMPLETED; RETURN CODE=nnn

Explanation: DB2STOP processing message.

User response: No action is required.

CKZ26507W ERROR CALLING CKZ01HEX; FUNCTION: function R15=nnnn

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

CKZ26508E UNABLE TO LOAD PROGRAM: program name

Explanation: The indicated program name was not found. Processing terminates.

User response: Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

CKZ26520E DSNALI ERROR; FUNCTION= function RC=nnnn RSN=nnnnnnnn

Explanation: An error occurred calling DSNALI. Function is the CAF function requested and RSN is the DB2 error reason. Processing terminates.

User response: Check the DB2 Messages and Codes manual for the RSN value. If unable to resolve the problem, contact IBM Software Support.

CKZ26521E IFI ERROR; FUNCTION= function RC=nnnn RSN=nnnnnnnn

Explanation: An error occurred using the DB2 IFI interface. Function is the IFI function requested and RSN is the DB2 error reason. Processing terminates.

User response: Check the DB2 Messages and Codes manual for the RSN value. If unable to resolve the problem, contact IBM Software Support.

CKZ26522E DB2 SUBSYSTEM: db2ssid NOT ACTIVE, RSN=00F30002 | DB2 SUBSYSTEM: db2ssid NOT DEFINED, RSN=00F30006

Explanation: The DB2 subsystem is not active or not defined to z/OS. Processing terminates.

User response: Verify that the correct DB2 subsystem ID is being used. If unable to resolve the problem,

contact IBM Software Support.

**CKZ26523I CONNECTED TO DB2 SUBSYSTEM:
ssid VERSION: nnn**

Explanation: A connection has been established to the DB2 subsystem. Processing continues.

User response: No action is required.

**CKZ26534E USERID IS NOT AUTHORIZED TO
ISSUE THE DB2 COMMAND**

Explanation: The userid running the job is not authorized to issue the DB2 command. Processing terminates.

User response: Verify the userid running the job has SYSADM authority.

CKZ26535E COMMAND FAILED; RC= nnnn

Explanation: The DB2 command failed. Processing terminates.

User response: Check the messages from the command failure and correct the problem. If unable to resolve the problem, contact IBM Software Support.

**CKZ26536E DB2 DID NOT TERMINATE WITHIN
WAIT TIME LIMIT**

Explanation: The DB2 subsystem did not complete its termination within the time limit specified in the WAIT keyword. Processing terminates.

User response: Determine why the DB2 subsystem did not complete its termination. If the wait time limit is too short, increase it.

CKZ26537I DB2 HAS TERMINATED

Explanation: The DB2 subsystem has terminated. Processing continues.

User response: No action is required.

CKZ26538I WAITING FOR DB2 TO TERMINATE

Explanation: DB2 Cloning Tool is waiting for the DB2 subsystem to terminate. Processing continues.

User response: No action is required.

CKZ26539I name IS STILL RUNNING

Explanation: The indicated DB2 address space is still running. Processing continues.

User response: No action is required.

CKZ26540I DB2 COMMAND: command

Explanation: Displays the DB2 command that will be issued. Processing continues.

User response: No action is required.

**CKZ26541I COMMAND NOT EXECUTED DUE TO
SIMULATION MODE**

Explanation: The DB2 command was not issued because this is a simulation run. Processing continues.

User response: No action is required.

**CKZ26542E UNABLE TO DETERMINE DB2
STATUS; DB2 IS RUNNING BUT IS
NOT ACCEPTING CONNECTIONS**

Explanation: DB2 Cloning Tool is unable to determine the status of the DB2 subsystem. The DB2 subsystem is running but is not accepting connections. Processing terminates.

User response: Get the DB2 subsystem either all the way up or stop it.

**CKZ26543I DB2 SUBSYSTEM TERMINATION
NOT COMPLETE**

Explanation: The DB2 subsystem has not completely terminated. Processing continues.

User response: No action is required.

**CKZ26548I OPTIONS IN EFFECT FOR THIS
EXECUTION: merged options**

Explanation: Informational message indicating how DB2STOP will handle the options. The displayed options are derived from the CKZINI and any overriding specifications in the command input.

User response: No action is required.

**CKZ26550E ERROR IN PARAMETERS FOR
keyword**

Explanation: The parameters for the indicated keyword were incorrect. Processing terminates.

User response: Check the keyword parameters. Mutually exclusive keywords may have been used.

**CKZ26551E REQUIRED KEYWORD MISSING:
keyword**

Explanation: A keyword required for processing has been omitted. Processing terminates.

User response: Specify the required keyword.

CKZ26553E KEYWORD: keyword MAXIMUM LENGTH: nnn EXCEEDED

Explanation: The operand entered for a keyword exceeded the maximum length allowed for the operand. nnn is the maximum allowed length for the keyword. Processing terminates.

User response: Correct the length of the keyword's operand.

CKZ26554E KEYWORD HAS MORE THAN 1 OPERAND; ONLY 1 ALLOWED: keyword

Explanation: Multiple operands were detected for a keyword; only one operand is permitted. Processing terminates.

User response: Correct the keyword to use one operand.

CKZ26556E NOTHING SPECIFIED FOR KEYWORD: keyword

Explanation: A keyword was entered without an appropriate operand. Processing terminates.

User response: Specify an appropriate operand for the keyword.

CKZ26558E INVALID VALUE IN KEYWORD: keyword VALUE: value error text

Explanation: The value in the keyword is invalid. 'error text' indicates the problem detected with the value. Processing terminates.

User response: Correct the value specified in the keyword.

CKZ26560E section NOT AS EXPECTED

Explanation: The section specified does not have the expected format. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ26570I STOPPING DB2 SUBSYSTEM: ssid

Explanation: The DB2 subsystem is being stopped. Processing continues.

User response: No action is required.

CKZ26586I VALIDATING KEYWORD: keyword

Explanation: Parsing is checking the indicated keyword indicated in the command.

User response: No action is required.

CKZ27001I hh:mm:ss DB2ALTERBSDS STARTED - PROGRAM REV=rrr | hh:mm:ss DB2ALTERBSDS COMPLETED; RETURN CODE=nnn

Explanation: DB2ALTERBDS processing message.

User response: No action is required.

CKZ27003I DDNAME=ddname ALLOCATED FOR DSN=datasetname

Explanation: 'ddname' has been dynamically allocated for the indicated data set.

User response: No action is required.

CKZ27004E DDNAME MISSING: ddname

Explanation: 'ddname' was specified for DB2 Cloning Tool to use. Processing terminates.

User response: Either correct the ddname specified, or, add the appropriate ddname to the job's JCL.

CKZ27005E ALLOCATION FAILED FOR DSN: datasetname

Explanation: Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ27005W DEALLOCATION FAILED FOR DDNAME: ddname

Explanation: Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ27007W ERROR CALLING CKZ01HEX; FUNCTION: function R15=nnnn

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

CKZ27008E UNABLE TO LOAD PROGRAM:
program name

Explanation: The indicated program name was not found. Processing terminates.

User response: Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

CKZ27009E ERROR ACCESSING JOURNAL FILE;
LOC=IIIII

Explanation: A VSAM error occurred accessing a file. Processing terminates.

User response: See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ27011E JOURNAL type RECORD NOT FOUND

Explanation: An expected record was not found in the DB2 Cloning Tool journal file. Processing terminates.

User response: Verify the same value was used for the DB2-NAME keyword that was used with the prior DB2UPDATE command for this DB2 subsystem or data sharing group. If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ27012E JOURNAL type RECORD IS WRONG
VERSION

Explanation: A journal record does not match the expected format. The record is printed. Processing terminates.

User response: Verify that different releases of DB2 Cloning Tool have not been run using the same journal data set. A journal created by a prior release of DB2 Cloning Tool can be upgraded to the current release by using the DB2 Cloning Tool JRNLUPGRADE command. See the "JRNLUPGRADE" on page 428 topic for the scenarios where JRNLUPGRADE can be used. If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ27013E COUNT MISMATCH; RECORDS
READ=rrrr CONTROL RECORD
COUNT=cccc LOC=IIIII

Explanation: There was a problem with the journal records needed to initiate the command. The number of records read from the journal, rrrr, is not the same as the number indicated in the journal DB2 control record, cccc. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ27020E NOTHING WAS REQUESTED TO BE
DONE

Explanation: No request keywords were included in the DB2ALTERBSDS command. Processing terminates.

User response: Add a request keyword to the DB2ALTERBSDS command.

CKZ27030I OPTIONS IN EFFECT FOR THIS
EXECUTION: merged options

Explanation: Informational message indicating how DB2ALTERBSDS will handle the options. The displayed options are derived from the CKZINI and any overriding specifications in the command input.

User response: No action is required.

CKZ27040E DB2UPDATE HAS NOT BEEN RUN
FOR THIS MEMBER

Explanation: DB2UPDATE has not been run for this member. Processing terminates.

User response: Run DB2UPDATE for the member.

CKZ27041E DB2UPDATE FOR THIS MEMBER
WAS SIMULATION, SIMULATION
MUST BE SPECIFIED

Explanation: DB2UPDATE for this member was run as a simulation and this run is not a simulation. Processing terminates.

User response: Run as SIMULATE or run DB2UPDATE for this member without SIMULATE.

CKZ27042E SLB-START WAS REQUESTED BUT
THERE WAS NO SLB INFORMATION
FOUND IN A BSDS

Explanation: SLB-START was requested, but no SLB information was found in a BSDS. Processing terminates.

User response: Verify that the BSDS should have SLB information in it. If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ27050E ERROR IN PARAMETERS FOR
keyword

Explanation: The parameters for the indicated keyword were incorrect. Processing terminates.

User response: Check the keyword parameters. Mutually exclusive keywords may have been used.

**CKZ27051E REQUIRED KEYWORD MISSING:
keyword**

Explanation: A keyword required for processing has been omitted. Processing terminates.

User response: Specify the required keyword.

**CKZ27052E REQUIRED KEYWORD NOT
ALLOWED: keyword FOR NON DATA
SHARING DB2**

Explanation: The DB2 system being updated is not part of a data group and a keyword that applies only to data sharing groups has been specified. Processing terminates.

User response: Remove the not allowed keyword from the command.

**CKZ27053E KEYWORD: keyword MAXIMUM
LENGTH: nnn EXCEEDED**

Explanation: The operand entered for a keyword exceeded the maximum length allowed for the operand. nnn is the maximum allowed length for the keyword. Processing terminates.

User response: Correct the length of the keyword's operand.

**CKZ27054E KEYWORD HAS MORE THAN 1
OPERAND; ONLY 1 ALLOWED:
keyword**

Explanation: Multiple operands were detected for a keyword; only one operand is permitted. Processing terminates.

User response: Correct the keyword to use one operand.

**CKZ27056E NOTHING SPECIFIED FOR
KEYWORD: keyword**

Explanation: A keyword was entered without an appropriate operand. Processing terminates.

User response: Specify an appropriate operand for the keyword.

**CKZ27058E INVALID VALUE IN KEYWORD:
keyword VALUE: value error text**

Explanation: The value in the keyword is invalid. 'error text' indicates the problem detected with the value. Processing terminates.

User response: Correct the value specified in the keyword.

CKZ27086I VALIDATING KEYWORD: keyword

Explanation: Parsing is checking the indicated keyword indicated in the command.

User response: No action is required.

**CKZ27101I hh:mm:ss BSDS ALTER STARTED -
PROGRAM REV=rrr | hh:mm:ss BSDS
ALTER COMPLETED; RETURN
CODE=nnn**

Explanation: BSDS ALTER processing message.

User response: No action is required.

**CKZ27103I DDNAME=ddname ALLOCATED FOR
DSN=datasetname**

Explanation: 'ddname' has been dynamically allocated for the indicated data set.

User response: No action is required.

CKZ27104E OPEN FAILED FOR DDNAME=ddname

Explanation: 'ddname' was allocated for DB2 Cloning Tool to use, but, the open for the file failed. Processing terminates.

User response: If unable to determine the reason the open failed, contact IBM Software Support. Have available the listing that contains this message.

**CKZ27105E ALLOCATION FAILED FOR DSN:
datasetname**

Explanation: Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ27105W DEALLOCATION FAILED FOR
DDNAME: ddname**

Explanation: Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

**CKZ27106E ERROR CALLING CKZ01VV1 tttttt
FUNCTION: function R15=nnnn
R0=nnnnnnnn LOC=llll**

Explanation: A problem occurred using a daspace. tttttt is the name of the internal table. llll is the location where the error occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

**CKZ27107W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn**

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

**CKZ27109E ERROR ACCESSING BSDS FILE;
LOC=llll**

Explanation: A VSAM error occurred accessing a file. Processing terminates.

User response: See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

| **CKZ27110E ERROR ACCESSING ACTIVE LOG;
| DDNAME: ddname R15=nnnn
| ERROR=nnn FUNCTION=function
| LOC=llll RBA OF RECORD:
| X'nnnnnnnn_nnnnnnn'**

| **Explanation:** A VSAM error occurred accessing the indicated active log file. Processing terminates.

| **User response:** If unable to resolve problem, contact IBM Software Support. Have available the listing that contains this message.

**CKZ27120E BSDS CONTROL RECORD NOT
FOUND**

Explanation: The control record was not found in the BSDS. The BSDS is not valid. Processing terminates.

User response: Determine why the BSDS became invalid. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains this message.

**CKZ27121I HIGHEST LOG RBA
WRITTEN=xxxxxxxxxx**

Explanation: Informational message showing the highest log rba written recorded in the BSDS.

User response: No action is required.

**CKZ27122E BSDS RECORD NOT FOUND FOR
recordtype**

Explanation: An expected BSDS record type was not found. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ27123I ACTIVE LOG COPY n
DSN=datasetname START RBA=xxx**

Explanation: This is an informational message that displays the data set name and start RBA of an active log.

User response: No action is required.

| **CKZ27124I ACTIVE LOG DOES NOT CONTAIN
| SLB LRSN; CONDITIONAL RESTART
| RECORD WILL NOT BE CREATED**

| **Explanation:** The active log of this DB2 subsystem does not contain the SLB LRSN being used for a conditional restart. A conditional restart record will not be created for this DB2 subsystem.

| **User response:** No action is required.

**CKZ27135E DDNAME=ddname IS ALREADY
ALLOCATED**

Explanation: The ddname is already allocated and it must not be.

User response: Ensure the ddname is not allocated in the JCL. If it is not in the JCL, contact IBM Software Support. Have available the listing that contains this message.

**CKZ27140E DSNJU003 UTILITY EXECUTION
FAILED**

Explanation: The invoked DSNJU003 utility has failed.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ27150W NO ARCHIVE LOGS TO DELETE

Explanation: No archive logs met the selection criteria for deletion from the BSDS.

User response: No action is required.

CKZ27151W NO ACTIVE LOGS TO DELETE

Explanation: No active logs met the selection criteria for deletion from the BSDS.

User response: No action is required.

CKZ27501I hh:mm:ss DB2SETLOG STARTED - PROGRAM REV=rrr | hh:mm:ss DB2SETLOG COMPLETED; RETURN CODE=nnn

Explanation: DB2SETLOG processing message.

User response: No action is required.

CKZ27507W ERROR CALLING CKZ01HEX; FUNCTION: function R15=nnnn

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

CKZ27508E UNABLE TO LOAD PROGRAM: program name

Explanation: The indicated program name was not found. Processing terminates.

User response: Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

CKZ27520E DSNALI ERROR; FUNCTION= function RC=nnnn RSN=nnnnnnnn

Explanation: An error occurred calling DSNALI. Function is the CAF function requested and RSN is the DB2 error reason. Processing terminates.

User response: Check the DB2 Messages and Codes manual for the RSN value. If unable to resolve the problem, contact IBM Software Support.

CKZ27521E IFI ERROR; FUNCTION= function RC=nnnn RSN=nnnnnnnn

Explanation: An error occurred using the DB2 IFI interface. Function is the IFI function requested and RSN is the DB2 error reason. Processing terminates.

User response: Check the DB2 Messages and Codes manual for the RSN value. If unable to resolve the problem, contact IBM Software Support.

CKZ27522E DB2 SUBSYSTEM: db2ssid NOT ACTIVE, RSN=00F30002 | DB2 SUBSYSTEM: db2ssid NOT DEFINED, RSN=00F30006

Explanation: The DB2 subsystem is not active or not defined to z/OS. Processing terminates.

User response: Verify that the correct DB2 subsystem ID is being used. If unable to resolve the problem, contact IBM Software Support.

CKZ27523I CONNECTED TO DB2 SUBSYSTEM: ssid VERSION: nnn

Explanation: A connection has been established to the DB2 subsystem. Processing continues.

User response: No action is required.

CKZ27534E USERID IS NOT AUTHORIZED TO ISSUE THE DB2 COMMAND

Explanation: The userid running the job is not authorized to issue the DB2 command. Processing terminates.

User response: Verify the userid running the job has SYSADM authority.

CKZ27535E COMMAND FAILED; RC= nnnn

Explanation: The DB2 command failed. Processing terminates.

User response: Check the messages from the command failure and correct the problem. If unable to resolve the problem, contact IBM Software Support.

CKZ27540I DB2 COMMAND: command

Explanation: Displays the DB2 command that will be issued. Processing continues.

User response: No action is required.

CKZ27541I COMMAND NOT EXECUTED DUE TO SIMULATION MODE

Explanation: The start command was not issued because this is a simulation run. Processing continues.

User response: No action is required.

CKZ27548I OPTIONS IN EFFECT FOR THIS EXECUTION: merged options

Explanation: Informational message indicating how DB2SETOG will handle the options. The displayed options are derived from the CKZINI and any overriding specifications in the command input.

User response: No action is required.

CKZ27551E REQUIRED KEYWORD MISSING: keyword

Explanation: A keyword required for processing has been omitted. Processing terminates.

User response: Specify the required keyword.

CKZ27553E KEYWORD: keyword MAXIMUM LENGTH: nnn EXCEEDED

Explanation: The operand entered for a keyword exceeded the maximum length allowed for the operand. nnn is the maximum allowed length for the keyword. Processing terminates.

User response: Correct the length of the keyword's operand.

CKZ27554E KEYWORD HAS MORE THAN 1 OPERAND; ONLY 1 ALLOWED: keyword

Explanation: Multiple operands were detected for a keyword; only one operand is permitted. Processing terminates.

User response: Correct the keyword to use one operand.

CKZ27556E NOTHING SPECIFIED FOR KEYWORD: keyword

Explanation: A keyword was entered without an appropriate operand. Processing terminates.

User response: Specify an appropriate operand for the keyword.

CKZ27570I SUSPENDING DB2 SUBSYSTEM: ssid | RESUMING DB2 SUBSYSTEM: ssid

Explanation: The DB2 subsystem is being suspended or resumed. Processing continues.

User response: No action is required.

CKZ40001I hh:mm:ss BCSCLEAN STARTED - PROGRAM REV=rrr | hh:mm:ss BCSCLEAN COMPLETED; RETURN CODE=nnn

Explanation: BCSCLEAN command processing message.

User response: No action is required.

CKZ40003I DDNAME=ddname ALLOCATED FOR DSN=datasetname

Explanation: 'ddname' has been dynamically allocated for the indicated data set.

User response: No action is required.

CKZ40004E DDNAME MISSING: ddname

Explanation: 'ddname' was specified for DB2 Cloning Tool to use. Processing terminates.

User response: Either correct the ddname specified, or, add the appropriate ddname to the job's JCL.

CKZ40005E ALLOCATION FAILED FOR DSN: datasetname

Explanation: Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ40005W DEALLOCATION FAILED FOR DDNAME: ddname

Explanation: Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ40008E UNABLE TO LOAD PROGRAM: program name

Explanation: The indicated program name was not found. Processing terminates.

User response: Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

CKZ40009E ERROR ACCESSING JOURNAL FILE; LOC=IIIII

Explanation: A VSAM error occurred accessing the journal file. Processing terminates.

User response: See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ40010E No dsnmask specified for CLEANUP-CATALOG-DSNMASKS – at least 1 dsnmask is required.

Explanation: The CLEANUP-CATALOG-DSNMASKS keyword requires at least one data set name mask.

User response: Supply at least one data set name mask for the keyword.

CKZ40048I OPTIONS IN EFFECT FOR THIS EXECUTION: merged options

Explanation: Informational message indicating how BCSCLEAN will handle the options. The displayed options are derived from the CKZINI and any overriding specifications in the command input.

User response: No action is required.

CKZ40051E REQUIRED KEYWORD MISSING:
keyword

Explanation: A keyword required for processing has been omitted. Processing terminates.

User response: Specify the required keyword.

**CKZ40052E REQUIRED INI SECTION/TOKEN MISSING: SECTION=section
TOKEN=token | REQUIRED INI VALUE MISSING FOR
SECTION=section TOKEN=token |
INVALID INI VALUE FOR
SECTION=section TOKEN=token**

Explanation: An error occurred validating the CKZINI member options. Processing terminates.

User response: Correct the CKZINI member.

CKZ40053E KEYWORD: keyword MAXIMUM LENGTH: nnn EXCEEDED

Explanation: The operand entered for a keyword exceeded the maximum length allowed for the operand. nnn is the maximum allowed length for the keyword. Processing terminates.

User response: Correct the length of the keyword's operand.

CKZ40054E KEYWORD HAS MORE THAN 1 OPERAND; ONLY 1 ALLOWED:
keyword

Explanation: Multiple operands were detected for a keyword; only one operand is permitted. Processing terminates.

User response: Correct the keyword to use one operand.

CKZ40055E Dsname mask has invalid characters at offset: offset mask: data_set_name_mask.

Explanation: Validation of the data set name mask failed. The error offset (starting with 0) is given in the message.

User response: Review the data set name masking requirements in the documentation, and make the necessary changes.

CKZ40056E NOTHING SPECIFIED FOR KEYWORD: keyword

Explanation: A keyword was entered without an appropriate operand. Processing terminates.

User response: Specify an appropriate operand for the keyword.

CKZ40057E Dsname mask too long: length data_set_name_mask.

Explanation: Data set name masks must be 44 characters or less. The data set name mask in error and its length are given in the message.

User response: Review the data set name masking requirements in the documentation, and make the necessary changes.

CKZ40058E INVALID VALUE IN KEYWORD: keyword VALUE: value error text

Explanation: The value in the keyword is invalid. 'error text' indicates the problem detected with the value. Processing terminates.

User response: Correct the value specified in the keyword.

CKZ41001I hh:mm:ss FINDUCATS STARTED - PROGRAM REV=rrr | hh:mm:ss FINDUCATS COMPLETED; RETURN CODE=nnn

Explanation: FINDUCATS processing message.

User response: No action is required.

CKZ41006E ERROR CALLING CKZ01VV1 tttttt FUNCTION: function R15=nnnn R0=nnnnnnnn LOC=lllll

Explanation: A problem occurred using a dataspace. tttttt is the name of the internal table. lllll is the location where the error occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

CKZ41008E UNABLE TO LOAD PROGRAM: program name

Explanation: The indicated program name was not found. Processing terminates.

User response: Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

CKZ41051E REQUIRED KEYWORD MISSING: keyword

Explanation: A keyword required for processing has been omitted. Processing terminates.

User response: Specify the required keyword.

CKZ41053E KEYWORD: keyword MAXIMUM LENGTH: nnn EXCEEDED

Explanation: The operand entered for a keyword exceeded the maximum length allowed for the operand. nnn is the maximum allowed length for the keyword. Processing terminates.

User response: Correct the length of the keyword's operand.

CKZ41054E KEYWORD HAS MORE THAN 1 OPERAND; ONLY 1 ALLOWED: keyword

Explanation: Multiple operands were detected for a keyword; only one operand is permitted. Processing terminates.

User response: Correct the keyword to use one operand.

CKZ41056E NOTHING SPECIFIED FOR KEYWORD: keyword

Explanation: A keyword was entered without an appropriate operand. Processing terminates.

User response: Specify an appropriate operand for the keyword.

CKZ41057E DUPLICATE FOUND; KEYWORD: keyword ENTRY: entry

Explanation: The indicated 'entry' for the keyword was previously specified. Processing terminates.

User response: Remove the duplicate entry.

CKZ41058E INVALID VALUE IN KEYWORD: keyword VALUE: value error text

Explanation: The value in the keyword is invalid. 'error text' indicates the problem detected with the value. Processing terminates.

User response: Correct the value specified in the keyword.

CKZ41060E UCBSCAN ERROR; RETURN CODE=nn REASON CODE=nn | UCBINFO ERROR; RETURN CODE=nn REASON CODE=nn

Explanation: An error occurred using UCBSCAN or UCBINFO. Processing terminates.

User response: Contact IBM Software Support. Have available the listing containing this message.

CKZ41061E CKZ01SMF ERROR; RETURN CODE=nnnn LOC: llllll entry

Explanation: An error occurred using CKZ01SMF to obtain SSI information for the 'entry'. llllll is the internal location where the error occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing containing this message.

CKZ41062E *Multiple messages and explanations; see Explanation*

Explanation:

Message

NO STORAGE GROUPS RETURNED BY SSI

Explanation

The FINDUCATS command was requested to determine volume serial numbers by storage group names. SSI did not return any storage groups to CKZ01SMF. Processing terminates.

Message

NO VOLUME SERIALS RETURNED BY SSI

Explanation

The FINDUCATS command was requested to check the SMS status of volume serials. SSI did not return any volume serials to CKZ01SMF. Processing terminates.

User response: Check that storage groups are defined on the system. If unable to determine the cause of this message, contact IBM Software Support. Have available the listing containing this message.

CKZ41063W NO VOL/STG MATCH FOUND; KEYWORD: keyword ENTRY: entry

Explanation: The indicated 'entry' for the keyword was not matched. For VOL, the indicated volser, or, the volser derived from a storage group, was not found. For STG, the indicated storage group was not found. Processing continues.

User response: No action is required.

CKZ41064E INVALID VOLSER: volser IN KEYWORD: keyword

Explanation: The volume serial number specified is invalid. Processing terminates.

User response: Correct the volser specification.

CKZ41067W NO VOLUME SERIALS FOR STORAGE GROUP: storage group

Explanation: SSI did not return any volsers for the storage group to CKZ01SMF. Processing continues.

User response: No action is required.

CKZ41075E NO VOLUME SERIALS SELECTED

Explanation: No volume serials have been found for FINDUCATS to examine. Processing terminates.

User response: Check if volsers specified in the FROM- keyword have been removed by use of an EXCLUDE-FROM- keyword.

CKZ41086I STORAGE GROUPS/MASKS FOR KEYWORD: keyword

Explanation: Parsing found the listed storage groups/masks for the keyword.

User response: No action is required.

CKZ41087I nnnnn VOLSERS OR MASKS FOR KEYWORD: keyword list of volsers | nnnnn VOLSERS DERIVED FOR KEYWORD: keyword list of volsers | nnnnn VOLSERS RESOLVED FOR KEYWORD: keyword

Explanation: Parsing found the listed volsers/masks for a keyword, or, derived the listed volsers from a storage group keyword. nnnnn is the number of volume serials. The third format, RESOLVED, indicates the number of volume serials found online for the associated keyword.

User response: No action is required.

CKZ41101I hh:mm:ss VOLUME COLLECTION STARTED - PROGRAM REV=rrr | hh:mm:ss VOLUME COLLECTION COMPLETED; RETURN CODE=nnn

Explanation: FINDUCATS volume processing message.

User response: No action is required.

CKZ41103I DDNAME=ddname ALLOCATED FOR DSN=datasetname

Explanation: 'ddname' has been dynamically allocated for the indicated data set.

User response: No action is required.

CKZ41104E OPEN FAILED FOR DDNAME=ddname

Explanation: 'ddname' was allocated for DB2 Cloning Tool to use, but, the open for the file failed. Processing terminates.

User response: If unable to determine the reason the open failed, contact IBM Software Support. Have available the listing that contains this message.

CKZ41105E ALLOCATION FAILED FOR DSN: datasetname

Explanation: Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ41105W DEALLOCATION FAILED FOR DDNAME: ddname

Explanation: Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ41106E ERROR CALLING CKZ01VV1 tttttt FUNCTION: function R15=nnnn R0=nnnnnnnn LOC=lllll

Explanation: A problem occurred using a dataspace. tttttt is the name of the internal table. lllll is the location where the error occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

CKZ41108E UNABLE TO LOAD PROGRAM: program name

Explanation: The indicated program name was not found. Processing terminates.

User response: Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

CKZ41130E AMSOPEN FAILED; R15=nnnn

Explanation: An attempt was made to issue an IDCAMS command. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ41131E NON-ZERO RETURNED BY IDCAMS; RC=nnnn

Explanation: An IDCAMS command failed with return code nnnn. The IDCAMS messages are displayed. Processing will terminate for an IDCAMS return code greater than 4.

User response: Check the volume that caused the DCOLLECT errors. Correct the problems with the volume.

CKZ41135I hh:mm:ss VOLSER volser processing step

Explanation: FINDUCATS volume processing message.

User response: No action is required.

CKZ41136I Report by volume serial numbers

Explanation: Indicates the start of the FINDUCATS report by volume serial number(s).

User response: No action is required.

CKZ41137I Report of user catalogs on all examined volume serials

Explanation: Indicates the start of the FINDUCATS report of catalogs.

User response: No action is required.

CKZ41141E CKZ00900 UNEXPECTED RESULTS; error text

Explanation: An unexpected condition occurred calling program CKZ00900. 'error text' has a description of the problem. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ41142E BAD MLA VALUE FOUND: mla value

Explanation: The MLA value found was invalid for DB2 Cloning Tool processing. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ41143E CSI STORAGE AREA WILL EXCEED 1MB

Explanation: Program IGGCSI00, Catalog Search Interface, required more storage to locate alias names. The amount of storage will exceed 1,048,575. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ41144E CSI RETURNED NO DATA

Explanation: This should not occur. Program IGGCSI00, Catalog Search Interface, did not return any data. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ41145W NO DATA SETS PROCESSED FROM VOLUME(S)

Explanation: No data sets other than SYS1.VTOCIX or SYS1.VVDS entries were found on the specified volumes. Processing continues.

User response: No action is required.

CKZ42001I hh:mm:ss VOLOPTIIONS STARTED - PROGRAM REV=rrr | hh:mm:ss VOLOPTIIONS COMPLETED; RETURN CODE=nnn

Explanation: VOLOPTIIONS processing message.

User response: No action is required.

CKZ42003I DDNAME=ddname ALLOCATED FOR DSN=datasetname

Explanation: 'ddname' has been dynamically allocated for the indicated data set.

User response: No action is required.

CKZ42004E DDNAME MISSING: ddname | OPEN FAILED FOR DDNAME: ddname

Explanation: 'ddname' was specified for DB2 Cloning Tool to use. Processing terminates.

User response: Either correct the ddname specified, or add the appropriate ddname to the job's JCL.

CKZ42005E ALLOCATION FAILED FOR DSN: datasetname

Explanation: Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ42005W DEALLOCATION FAILED FOR DDNAME: ddname

Explanation: Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ42006E ERROR CALLING CKZ01VV1 tttttt
FUNCTION: function R15=nnnn
R0=nnnnnnnn LOC=llll

Explanation: A problem occurred using a daspace. tttttt is the name of the internal table. llll is the location where the error occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

CKZ42007W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

CKZ42008E UNABLE TO LOAD PROGRAM:
program name

Explanation: The indicated program name was not found. Processing terminates.

User response: Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

CKZ42009E ERROR ACCESSING JOURNAL FILE;
LOC=llll

Explanation: A VSAM error occurred accessing the journal file. Processing terminates.

User response: See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ42010E DUPLICATE JOURNAL ENTRY;
LOC=llll

Explanation: A duplicate record was detected. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ42011E JOURNAL CONTROL RECORD NOT
FOUND | JOURNAL VOLUME PAIR
RECORD(S) NOT FOUND

Explanation: An expected record was not found in the DB2 Cloning Tool journal file. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ42012E JOURNAL CONTROL RECORD IS
WRONG VERSION | JOURNAL VOLP
RECORD IS WRONG VERSION

Explanation: The journal record does not match the expected format. The record is printed. Processing terminates.

User response: Verify that different releases of DB2 Cloning Tool have not been run using the same journal data set. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ42013E RECORD COUNT IS ZERO; LOC=llll
| COUNT MISMATCH; RECORDS
READ=rrrr CONTROL RECORD
COUNT=cccc

Explanation: There was a problem with the journal records needed to initiate the command. For the first format, the journal control record indicates no entries were added. For the second format, the number of records read from the journal, rrrr, is not the same as the number indicated in the journal control record, cccc. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ42015E THE COPY PROCESS DID NOT
COMPLETE SUCCESSFULLY

Explanation: The journal indicates that the COPY command did not complete successfully. Processing terminates.

User response: Check that the COPY command has completed successfully before initiating the VOLOPTIONS command.

CKZ42017E THE NEWTARGETS-DDN IS EMPTY
OR HAS BEEN DUMMIED, DDNAME:
ddn

Explanation: No records were read from the ddname specified for CKZMSG NEWTARGETS-DDN. Processing terminates.

User response: Check that the DD has not been specified as 'DD DUMMY' or 'DD DSN=NULLFILE'. Check that the DSN specified in the ddn has been created successfully.

CKZ42020I CURRENT JOURNAL VOLUME PAIRS:
| UPDATED JOURNAL VOLUME
PAIRS: (SIMULATION **) list of**
volume pairs

Explanation: The first format indicates the volume pairs at the initiation of the command. The second format indicates the volume pairs after the UPDATE NEWTARGETS have been applied.

User response: No action is required.

CKZ42021I UPDATED USERCATALOGS VOLUME:
(** SIMULATION **) list of usercatalogs

Explanation: Displays the usercatalogs and their volsers after the UPDATE NEWTARGETS have been applied.

User response: No action is required.

CKZ42031I VOLUMES FOR NEW TARGETS:
source target newtarget

Explanation: The listed groups of volume serials were obtained from the NEWTARGETS keyword, or, from the NEWTARGET-DDN file.

User response: No action is required.

CKZ42040E UCBLOOK ERROR; RETURN
CODE=nn REASON CODE=nn
LOC=lllll

Explanation: An error occurred during UCBLOOK processing. Processing terminates.

User response: Contact IBM Software Support. Have available the listing containing this message.

CKZ42051E REQUIRED KEYWORD MISSING:
keyword

Explanation: A keyword required for processing has been omitted. Processing terminates.

User response: Specify the required keyword.

CKZ42053E KEYWORD: keyword MAXIMUM
LENGTH: nnn EXCEEDED

Explanation: The operand entered for a keyword exceeded the maximum length allowed for the operand. nnn is the maximum allowed length for the keyword. Processing terminates.

User response: Correct the length of the keyword's operand.

CKZ42054E KEYWORD HAS MORE THAN 1
OPERAND; ONLY 1 ALLOWED:
keyword

Explanation: Multiple operands were detected for a keyword; only one operand is permitted. Processing terminates.

User response: Correct the keyword to use one operand.

CKZ42056E NOTHING SPECIFIED FOR
KEYWORD: keyword

Explanation: A keyword was entered without an appropriate operand. Processing terminates.

User response: Specify an appropriate operand for the keyword.

CKZ42057E DUPLICATE FOUND; KEYWORD:
keyword ENTRY: entry

Explanation: The indicated 'entry' for the keyword was previously specified. Processing terminates.

User response: Remove the duplicate entry.

CKZ42058E INVALID VALUE IN KEYWORD:
keyword VALUE: value error text

Explanation: The value in the keyword is invalid. 'error text' indicates the problem detected with the value. Processing terminates.

User response: Correct the value specified in the keyword.

CKZ42060E RESUME ONLY VALID WITH CLIP,
OFFLINECLIP, OR UNCLIP

Explanation: The RESUME keyword can only be specified if the CLIP, OFFLINECLIP, or UNCLIP keyword is specified. Processing terminates.

User response: Correct the keyword specification.

CKZ42063E INVALID DEVN: devn IN KEYWORD:
keyword

Explanation: The volume device number specified is invalid. Processing terminates.

User response: Correct the device number specification.

CKZ42064E INVALID VOLSER: volser IN
KEYWORD: keyword

Explanation: The volume serial number specified is invalid. Processing terminates.

User response: Correct the volser specification.

CKZ42066E VOLSER REFERENCED AS BOTH
SOURCE AND TARGET: volser

Explanation: The indicated new target volume serial is already in use as a source volume serial. Processing terminates.

User response: Correct the volume serial specification.

**CKZ42068E UNMATCHED ENTRIES IN
KEYWORD: keyword**

Explanation: For NEWTARGETS, there must be a source volume serial, target volume serial, new target volume serial. Unmatched entries were found. Processing terminates.

User response: Correct the keyword specification.

**CKZ42070E NO MATCH FOUND IN JOURNAL
FOR SOURCE VOLSER: volser**

Explanation: The indicated volser was specified as a source volume serial in the NEWTARGETS keyword. That source volume serial was not found in the journal records. Processing terminates.

User response: Correct the keyword specification. The VOLOPTIONS LIST command may be used to display the volume pairs in the journal.

**CKZ42071E SPECIFIED TARGET DOES NOT
MATCH JOURNAL ENTRY, TARGET:
volser**

Explanation: The indicated volser was specified as a target volume serial in the NEWTARGETS keyword. That target volume serial was not found paired to the specified source volume serial. Processing terminates.

User response: Correct the keyword specification. The VOLOPTIONS LIST command may be used to display the volume pairs in the journal.

**CKZ42072E SPECIFIED TARGET WILL CAUSE
DUPLICATE TARGET VOLUME
SERIALS, TARGET: volser**

Explanation: The indicated volser was specified as a target volume serial in the NEWTARGETS keyword. It is either duplicated in the new target volume serials specified, or, will duplicate an existing target volume serial that is not being changed. Processing terminates.

User response: Correct the keyword specification. The VOLOPTIONS LIST command may be used to display the volume pairs in the journal.

**CKZ42073E SPECIFIED TARGET WILL CAUSE
DUPLICATE TARGET VOLUME
DEVNS, TARGET DEVN: devn**

Explanation: The indicated device number was specified as a target volume device number in the NEWTARGETSDEVN keyword. It is either duplicated in the new target volume device numbers specified, or, will duplicate an existing target volume device number that is not being changed. Processing terminates.

User response: Correct the keyword specification. The VOLOPTIONS LIST command may be used to display the volume pairs in the journal.

**CKZ42101I hh:mm:ss VOLOPTIONS CLIP
STARTED - PROGRAM REV=rrr (**
SIMULATION **) | hh:mm:ss
VOLOPTIONS CLIP COMPLETED;
RETURN CODE=nnn**

Explanation: VOLOPTIONS CLIP processing message.

User response: No action is required.

**CKZ42107W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn**

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

**CKZ42108E UNABLE TO LOAD PROGRAM:
program name**

Explanation: The indicated program name was not found. Processing terminates.

User response: Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

**CKZ42109E ERROR ACCESSING JOURNAL FILE;
LOC=IIIII**

Explanation: A VSAM error occurred accessing the journal file. Processing terminates.

User response: See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

**CKZ42111E JOURNAL VOLUME PAIR RECORD(S)
NOT FOUND**

Explanation: An expected record was not found in the DB2 Cloning Tool journal file. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ42136E IEEVARYD VARY OFFLINE FAILED
FOR VOLSER: volser INVALID
PARAMETERS**

Explanation: The parameters given to IEEVARYD are incorrect. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ42137E IEEVARYD VARY OFFLINE FAILED
FOR VOLSER: volser DEVICE: nnnn
RETURN CODE: nnnnnnnn REASON
CODE: nnnnnnnn

Explanation: The vary offline for the indicated device failed. Processing terminates.

User response: When the problem that caused the vary to fail is corrected, any volumes previously varied offline will need to be varied back online in order to rerun with VOLOPTIONS CLIP.

CKZ42138E IEEVARYD VARY OFFLINE FAILED
FOR VOLSER: volser DEVICE: nnnn
R15: nnnnnnnn

Explanation: The vary offline for the indicated device failed. Processing terminates.

User response: When the problem that caused the vary to fail is corrected, any volumes previously varied offline will need to be varied back online in order to rerun with VOLOPTIONS CLIP.

CKZ42141I VOLUME SERIAL: volser DEVICE
NUMBER: nnnn IS NOW OFFLINE

Explanation: The indicated device is now offline to the current image.

User response: No action is required.

CKZ42142I VOLUME SERIAL: volser DEVICE
NUMBER: nnnn IS PENDING OFFLINE

Explanation: The indicated device is now pending offline to the current image.

User response: No action is required.

CKZ42143E SOURCE VOLUME SERIAL: volser IS
OFFLINE AND NO INFORMATION
ABOUT THE DEVICE IS KNOWN

Explanation: The indicated source volume serial is not online and no information about the physical device is known. Processing terminates.

User response: Bring the target device with the source volume serial online and run VOLOPTIONS CLIP with the RESUME keyword.

CKZ42144E DEVICE NUMBER: devn FOR TARGET
VOLUME: volser DOES NOT EXIST

Explanation: The indicated device number is not defined to the z/OS system. Processing terminates.

User response: Correct the device number to use a defined device.

CKZ42145E DEVICE NUMBER: devn IS ONLINE
WITH UNEXPECTED VOLUME
SERIAL: volser

Explanation: The indicated device number is online but the volser of the device does not match what is expected. Processing terminates.

User response: Either correct the device number to use the correct device or correct the device to have the correct contents.

CKZ42146E DEVICE NUMBER: devn IS ONLINE;
VOLUME SERIAL: volser

Explanation: The indicated device is online and OFFLINECLIP was specified. For OFFLINECLIP the target device should be offline. Processing terminates.

User response: Verify OFFLINECLIP is really desired. If this is a rerun of an OFFLINECLIP the RESUME keyword should be used.

CKZ42147I VOLUME SERIAL: volser DEVICE
NUMBER: devn NOT TAKEN OFFLINE
DUE TO SIMULATION

Explanation: The indicated volume was not taken offline because this run is a simulation.

User response: No action is required.

CKZ42148E SOURCE VOLUME SERIAL= vvvvvv IS
OFFLINE AND IS EXPECTED TO BE
ONLINE

Explanation: The indicated volume is offline but is expected to be online. Processing terminates.

User response: If this is a rerun, the RESUME keyword should be used.

CKZ42160E ERROR DURING UCBxxx FOR
VOLSER=volume - RETURN
CODE=nnnn REASON CODE=nnnn

Explanation: An error occurred using UCBLOOK or UCBSCAN. Processing terminates.

User response: A return code 4 from UCBLOOK may indicate the volser is offline. If unable to determine the cause of the error, contact IBM Software Support. Have available the listing containing this message.

CKZ42179W TARGET VOLUME SERIAL: volume IS
CURRENTLY ONLINE

Explanation: CLIP was requested, but, the indicated target volume serial was found online. Processing will continue checking all other targets, but, the run will be terminated.

User response: Change the target volume serial(s) DB2

Cloning Tool will use with the VOLOPTIONS UPDATE command to avoid duplicate volume serials.

CKZ42180E NO TARGET VOLSER SHOULD BE FOUND ONLINE FOR CLIP FUNCTION

Explanation: CLIP was requested, but, one or more target volumes were found online. Processing terminates.

User response: See the CKZ42179W message(s) for online target volume serials. If this is a rerun, the RESUME keyword should be used.

CKZ42201I hh:mm:ss VOLOPTIONS UNCLIP STARTED - PROGRAM REV=rrr (SIMULATION **) | hh:mm:ss VOLOPTIONS UNCLIP COMPLETED; RETURN CODE=nnn**

Explanation: VOLOPTIONS UNCLIP processing message.

User response: No action is required.

CKZ42203I DDNAME=ddname ALLOCATED FOR DSN=datasetname

Explanation: 'ddname' has been dynamically allocated for the indicated data set.

User response: No action is required.

CKZ42204E OPEN FAILED FOR DDNAME: ddname

Explanation: 'ddname' was specified for DB2 Cloning Tool to use. Processing terminates.

User response: Either correct the ddname specified, or add the appropriate ddname to the job's JCL.

CKZ42205E ALLOCATION FAILED FOR DSN: datasetname

Explanation: Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ42205W DEALLOCATION FAILED FOR DDNAME: ddname

Explanation: Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact

IBM Software Support. Have available the listing containing these messages.

CKZ42207W ERROR CALLING CKZ01HEX; FUNCTION: function R15=nnnn

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

CKZ42209E ERROR ACCESSING JOURNAL FILE; LOC=IIIII

Explanation: A VSAM error occurred accessing the journal file. Processing terminates.

User response: See associated CKZVSEnnE error messages. If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ42211E JOURNAL VOLUME PAIR RECORD(S) NOT FOUND

Explanation: An expected record was not found in the DB2 Cloning Tool journal file. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ42220E INTERNAL ERROR; INVALID REQUEST TYPE= type

Explanation: An internal error has been encountered. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ42235E ICKDSF COMMAND FAILED FOR TARGET VOLSER: volser

Explanation: The invocation of ICKDSF to change a device label failed. The messages from ICKDSF are printed.

User response: If unable to determine the reason for the failure from the associated ICKDSF messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ42236E IEEVARYD VARY OFFLINE FAILED FOR VOLSER: volser INVALID PARAMETERS

Explanation: The parameters given to IEEVARYD are incorrect. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ42237E IEEVARYD VARY OFFLINE FAILED
FOR VOLSER: volser DEVICE: nnnn
RETURN CODE: nnnnnnnn REASON
CODE: nnnnnnnn

Explanation: The vary offline for the indicated device failed. Processing terminates.

User response: When the problem that caused the vary to fail is corrected, the VOLOPTIONS UNCLIP can be rerun with the RESUME keyword.

CKZ42238E IEEVARYD VARY OFFLINE FAILED
FOR VOLSER: volser DEVICE: nnnn
R15: nnnnnnnn

Explanation: The vary offline for the indicated device failed. Processing terminates.

User response: When the problem that caused the vary to fail is corrected, the VOLOPTIONS UNCLIP can be rerun with the RESUME keyword.

CKZ42240I DEVICE NUMBER: nnnn
SUCCESSFULLY CHANGED TO
VOLUME SERIAL: volser

Explanation: The indicated device has been clipped to the indicated volume serial.

User response: No action is required.

CKZ42241I VOLUME SERIAL: volser DEVICE
NUMBER: nnnn IS NOW OFFLINE |
VOLUME SERIAL: volser DEVICE
NUMBER: nnnn IS ALREADY
OFFLINE

Explanation: The indicated device is now offline or was already offline to the current image.

User response: No action is required.

CKZ42242I VOLUME SERIAL: volser DEVICE
NUMBER: nnnn IS PENDING OFFLINE

Explanation: The indicated device is now pending offline to the current image.

User response: No action is required.

CKZ42243E TARGET VOLUME SERIAL: volser IS
OFFLINE AND NO INFORMATION
ABOUT THE DEVICE IS KNOWN

Explanation: The indicated target volume serial is not online and no information about the physical device is known. Processing terminates.

User response: Bring the target device with the target volume serial online and the VOLOPTIONS UNCLIP can be rerun with the RESUME keyword.

CKZ42244E TARGET VOLUME SERIAL= vvvvvv IS
OFFLINE AND IS EXPECTED TO BE
ONLINE

Explanation: The indicated volume is offline but is expected to be online. Processing terminates.

User response: If this is a rerun, the RESUME keyword should be used.

CKZ42245I DEVICE NUMBER: devn ALREADY
CHANGED TO SOURCE VOLUME
SERIAL: volser

Explanation: The indicated device number has already been clipped to the desired volume serial.

User response: No action is required.

CKZ42246I DEVICE NUMBER: nnnn ICKDSF
FAILED; DEVICE MAY HAVE
ALREADY BEEN UNCLIPPED

Explanation: The unclip of the indicated device by ICKDSF has failed. This failure is probably caused by the device having already been unclipped.

User response: See the next DB2 Cloning Tool message in the listing to determine the appropriate action.

CKZ42247I VOLUME SERIAL: volser DEVICE
NUMBER: devn NOT TAKEN OFFLINE
DUE TO SIMULATION

Explanation: The indicated volume was not taken offline because this run is a simulation.

User response: No action is required.

CKZ42248I VOLUME SERIAL: vvvvvv DEVICE
NUMBER: nnnn NOT CLIPPED TO
VOLUME SERIAL: vvvvvv DUE TO
SIMULATION

Explanation: The clip of the indicated device was not done because this run is a simulation.

User response: No action is required.

CKZ42260E ERROR DURING UCBxxx FOR
VOLSER=volume - RETURN
CODE=nnnn REASON CODE=nnnn

Explanation: An error occurred using UCBLOOK or UCBSCAN. Processing terminates.

User response: A return code 4 from UCBLOOK may indicate the volser is offline. If unable to determine the cause of the error, contact IBM Software Support. Have available the listing containing this message.

CKZ43001I hh:mm:ss UCATOPTIONS STARTED -
PROGRAM REV=rrr | hh:mm:ss
UCATOPTIONS COMPLETED;
RETURN CODE=nnn

Explanation: UCATOPTIONS processing message.

User response: No action is required.

CKZ43003I DDNAME=ddname ALLOCATED FOR
DSN=datasetname

Explanation: 'ddname' has been dynamically allocated for the indicated data set.

User response: No action is required.

CKZ43004E DDNAME MISSING: ddname | OPEN
FAILED FOR DDNAME: ddname

Explanation: 'ddname' was specified for DB2 Cloning Tool to use. Processing terminates.

User response: Either correct the ddname specified, or, add the appropriate ddname to the job's JCL.

CKZ43004E DDNAME MISSING: ddname | OPEN
FAILED FOR DDNAME: ddname

Explanation: 'ddname' was specified for DB2 Cloning Tool to use. Processing terminates.

User response: Either correct the ddname specified, or, add the appropriate ddname to the job's JCL.

CKZ43005E ALLOCATION FAILED FOR DSN:
datasetname

Explanation: Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ43005W DEALLOCATION FAILED FOR
DDNAME: ddname

Explanation: Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ43006E ERROR CALLING CKZ01VV1 tttttt
FUNCTION: function R15=nnnn
R0=nnnnnnnn LOC=lllll

Explanation: A problem occurred using a dataspace. tttttt is the name of the internal table. llll is the location where the error occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

CKZ43007W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

CKZ43008E UNABLE TO LOAD PROGRAM:
program name

Explanation: The indicated program name was not found. Processing terminates.

User response: Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

CKZ43009E ERROR ACCESSING JOURNAL FILE;
LOC=lllll

Explanation: A VSAM error occurred accessing the journal file. Processing terminates.

User response: See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ43010E DUPLICATE JOURNAL ENTRY;
LOC=lllll

Explanation: A duplicate record was detected. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ43011E JOURNAL CONTROL RECORD NOT
FOUND | JOURNAL UCAT PAIR
RECORD(S) NOT FOUND

Explanation: An expected record was not found in the DB2 Cloning Tool journal file. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ43012E JOURNAL CONTROL RECORD IS WRONG VERSION | JOURNAL UCAT PAIR RECORD IS WRONG VERSION

Explanation: The journal record does not match the expected format. The record is printed. Processing terminates.

User response: Verify that different releases of DB2 Cloning Tool have not been run using the same journal data set. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ43013E RECORD COUNT IS ZERO; LOC=IIIII | COUNT MISMATCH; RECORDS READ=rrrr CONTROL RECORD COUNT=cccc

Explanation: There was a problem with the journal records needed to initiate the command. For the first format, the journal control record indicates no entries were added. For the second format, the number of records read from the journal, rrrr, is not the same as the number indicated in the journal control record, cccc. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ43013E RECORD COUNT IS ZERO; LOC=IIIII | COUNT MISMATCH; RECORDS READ=rrrr CONTROL RECORD COUNT=cccc

Explanation: There was a problem with the journal records needed to initiate the command. For the first format, the journal control record indicates no entries were added. For the second format, the number of records read from the journal, rrrr, is not the same as the number indicated in the journal control record, cccc. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ43015E THE COPY PROCESS DID NOT COMPLETE SUCCESSFULLY

Explanation: The journal indicates that the COPY command did not complete successfully. Processing terminates.

User response: Check that the COPY command has completed successfully before initiating the VOLOPTIONS command.

CKZ43017E THE NEWTARGETS-DDN IS EMPTY OR HAS BEEN DUMMIED, DDNAME: ddn

Explanation: No records were read from the ddname

specified for NEWTARGETS-DDN. Processing terminates.

User response: Check that the DD has not been specified as 'DD DUMMY' or 'DD DSN=NULLFILE'. Check that the DSN specified in the ddn has been created successfully.

CKZ43018E THE NEWTARGETS-DDN INPUT HAS EXCEEDED THE CURRENT CAPACITY

Explanation: The number of entries read from the ddname exceeded the current capacity. Processing terminates.

User response: The UCATOPTIONS UPDATE may be run multiple times with the input split into smaller parts. In addition, please report this message to IBM Software Support.

CKZ43020I CURRENT JOURNAL UCAT PAIRS: | UPDATED JOURNAL UCAT PAIRS:

Explanation: The first format indicates the user catalog pairs at the initiation of the command. The second format indicates the user catalog pairs after the UPDATE NEWTARGETS have been applied.

User response: No action is required.

CKZ43031I ENTRIES FROM NEWTARGETS: source newtarget | ENTRIES FROM NEWCATWORKS: current-dsn new-dsn

Explanation: The listed pairs of catalog names were obtained from the NEWTARGETS keyword, or, from the NEWTARGETS-DDN file. Or, the listed pairs of data set names were obtained from the NEWCATWORKS keyword, or, from the NEWCATWORKS-DDN file.

User response: No action is required.

CKZ43040E SOURCE CATALOG BACKUP HAS NOT BEEN DONE

Explanation: The source catalogs have not been backed up. Processing terminates.

User response: Run UCATOPTIONS BACKUP to backup the source catalogs.

CKZ43041E SOURCE CATALOG BACKUP HAS ALREADY BEEN DONE

Explanation: The source catalogs have already been backed up. Processing terminates.

User response: UCATOPTIONS BACKUP does not need to be run.

CKZ43048I OPTIONS IN EFFECT FOR THIS EXECUTION: merged options

Explanation: Informational message indicating how UCATOPTIONS will handle the options. The displayed options are derived from the CKZINI and any overriding specifications in the command input.

User response: No action is required.

CKZ43051E REQUIRED KEYWORD MISSING: keyword

Explanation: A keyword required for processing has been omitted. Processing terminates.

User response: Specify the required keyword.

CKZ43052E REQUIRED INI SECTION/TOKEN MISSING: SECTION=section TOKEN=token | REQUIRED INI VALUE MISSING FOR SECTION=section TOKEN=token | INVALID INI VALUE FOR SECTION=section TOKEN=token

Explanation: An error occurred validating the CKZINI member options. Processing terminates.

User response: Correct the CKZINI member.

CKZ43053E KEYWORD: keyword MAXIMUM LENGTH: nnn EXCEEDED

Explanation: The operand entered for a keyword exceeded the maximum length allowed for the operand. nnn is the maximum allowed length for the keyword. Processing terminates.

User response: Correct the length of the keyword's operand.

CKZ43054E KEYWORD HAS MORE THAN 1 OPERAND; ONLY 1 ALLOWED: keyword

Explanation: Multiple operands were detected for a keyword; only one operand is permitted. Processing terminates.

User response: Correct the keyword to use one operand.

CKZ43056E NOTHING SPECIFIED FOR KEYWORD: keyword

Explanation: A keyword was entered without an appropriate operand. Processing terminates.

User response: Specify an appropriate operand for the keyword.

CKZ43057E DUPLICATE FOUND; KEYWORD: keyword ENTRY: entry

Explanation: The indicated 'entry' for the keyword was previously specified. Processing terminates.

User response: Remove the duplicate entry.

CKZ43058E INVALID VALUE IN KEYWORD: keyword VALUE: value error text

Explanation: The value in the keyword is invalid. 'error text' indicates the problem detected with the value. Processing terminates.

User response: Correct the value specified in the keyword.

CKZ43064E INVALID UCAT: bcs dsname IN KEYWORD: keyword | INVALID CATWORK: dsname IN KEYWORD: keyword

Explanation: The indicated bcs dsname or dsname is invalid. Processing terminates.

User response: Correct the invalid dsname.

CKZ43065E FORCE CAN ONLY BE SPECIFIED WITH BACKUP

Explanation: The FORCE keyword was specified, but the BACKUP keyword was not specified. The FORCE keyword can only be specified with the BACKUP keyword. Processing terminates.

User response: Correct the keyword specification.

CKZ43068E UNMATCHED ENTRIES IN KEYWORD: keyword

Explanation: For NEWTARGETS, there must be a source user catalog and a new target user catalog. Unmatched entries were found. Processing terminates.

User response: Correct the keyword specification.

CKZ43073E NO MATCH FOUND IN JOURNAL FOR SOURCE UCAT: codename | NO MATCH FOUND IN JOURNAL FOR CATWORK DSN: dsname

Explanation: The indicated user catalog was specified as a source in the NEWTARGETS keyword. That source usercatalog was not found in the journal records. Or, the indicated dsname was specified as a current value in the NEWCATWORKS keyword. That catwork dsname was not found in the journal records. Processing terminates.

User response: Correct the keyword specification. The UCATOPTIONS LIST command may be used to display the ucat pairs in the journal.

CKZ43074W NEWTARGETS IGNORED WITH "LIST" OPTION | NEWCATWORKS IGNORED WITH "LIST" OPTION

Explanation: NEWTARGETS or NEWCATWORKS was specified with UCATOPTIONS LIST. The NEWTARGETS or NEWCATWORKS keyword is ignored. Processing continues.

User response: None, unless UPDATE was intended.

CKZ44001I hh:mm:ss ONLINECLIP STARTED - PROGRAM REV=rrr | hh:mm:ss ONLINECLIP COMPLETED; RETURN CODE=nnn

Explanation: ONLINECLIP command processing message.

User response: No action is required.

CKZ44003I DDNAME=ddname ALLOCATED FOR DSN=datasetname

Explanation: 'ddname' has been dynamically allocated for the indicated data set.

User response: No action is required.

CKZ44004E DDNAME MISSING: ddname

Explanation: 'ddname' was specified for DB2 Cloning Tool to use. Processing terminates.

User response: Either correct the ddname specified, or add the appropriate ddname to the job's JCL.

CKZ44005E ALLOCATION FAILED FOR DSN: datasetname

Explanation: Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ44005W DEALLOCATION FAILED FOR DDNAME: ddname

Explanation: Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ44007W ERROR CALLING CKZ01HEX; FUNCTION: function R15=nnnn

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

CKZ44009E ERROR ACCESSING JOURNAL FILE; LOC=IIIII

Explanation: A VSAM error occurred accessing the journal file. Processing terminates.

User response: See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ44011E JOURNAL CONTROL RECORD NOT FOUND | JOURNAL CONTROL RECORD IS WRONG VERSION | JOURNAL VOLUME PAIR RECORD(S) NOT FOUND

Explanation: An expected journal record was not found or did not match the expected format. If the format is the problem, the record is printed. Processing terminates.

User response: Verify that different releases of DB2 Cloning Tool have not been run using the same journal data set. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ44013E RECORD COUNT IS ZERO; LOC=IIIII | COUNT MISMATCH; RECORDS READ=rrrr CONTROL RECORD COUNT=cccc

Explanation: There was a problem with the journal records needed to initiate the command. For the first format, the journal control record indicates no entries were added. For the second format, the number of records read from the journal, rrrr, is not the same as the number indicated in the journal control record, cccc. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ44015E THE COPY PROCESS DID NOT COMPLETE SUCCESSFULLY

Explanation: The journal indicates that the COPY command did not complete successfully. Processing terminates.

User response: Check that the COPY command has completed successfully before initiating the ONLINECLIP command.

CKZ44033E EXCP FAILED FOR DEVICE: nnnn
 TARGET VOLSER: volser SYNAD
 TEXT: text

Explanation: An error occurred accessing the volume label for a target device. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ44034W VOLUME SERIAL FOR DEVICE: nnnn
 IS volser; VOLUME SERIAL
 EXPECTED IS sourcevolser

Explanation: The internal volume label for the device is 'volser'. The program expected the internal label to match the associated source volume serial. Processing continues.

User response: If the internal volume serial is already the target volume serial, there should be no problem. However, if the internal serial number is not related to the current DB2 Cloning Tool process, check that the volume pairs given to the DB2 Cloning Tool COPY command were correct.

CKZ44035I VOLUME SERIAL FOR DEVICE: nnnn
 CHANGED TO: targetvolser

Explanation: The internal volume label for the device has been changed to the target volume serial.

User response: No action is required.

CKZ44048I OPTIONS IN EFFECT FOR THIS
 EXECUTION: merged options

Explanation: Informational message indicating how ONLINECLIP will handle the options. The displayed options are derived from the CKZINI and any overriding specifications in the command input.

User response: No action is required.

CKZ44050E ERROR IN PARAMETERS FOR
 keyword

Explanation: The parameters for the indicated keyword were incorrect. Processing terminates.

User response: Check the keyword parameters. Mutually exclusive keywords may have been used.

CKZ44051E REQUIRED KEYWORD MISSING:
 keyword

Explanation: A keyword required for processing has been omitted. Processing terminates.

User response: Specify the required keyword.

CKZ44053E KEYWORD: keyword MAXIMUM
 LENGTH: nnn EXCEEDED

Explanation: The operand entered for a keyword exceeded the maximum length allowed for the operand. nnn is the maximum allowed length for the keyword. Processing terminates.

User response: Correct the length of the keyword's operand.

CKZ44054E KEYWORD HAS MORE THAN 1
 OPERAND; ONLY 1 ALLOWED:
 keyword

Explanation: Multiple operands were detected for a keyword; only one operand is permitted. Processing terminates.

User response: Correct the keyword to use one operand.

CKZ44056E NOTHING SPECIFIED FOR
 KEYWORD: keyword

Explanation: A keyword was entered without an appropriate operand. Processing terminates.

User response: Specify an appropriate operand for the keyword.

CKZ44058E INVALID VALUE IN KEYWORD:
 keyword VALUE: value error text

Explanation: The value in the keyword is invalid. 'error text' indicates the problem detected with the value. Processing terminates.

User response: Correct the value specified in the keyword.

CKZ44060E ERROR DURING function FOR
 VOLSER=volume - RETURN
 CODE=nnnn REASON CODE=nnnn

Explanation: An error occurred using IOSCAPU or UCBLOOK. Processing terminates.

User response: A return code 4 from UCBLOOK may indicate the volser is offline. If unable to determine the cause of the error, contact IBM Software Support. Have available the listing containing this message.

CKZ44079E TARGET VOLSER: targetvolser WAS
 NOT FOUND ONLINE

Explanation: The indicated target volume serial was not found online. Processing terminates.

User response: The target volumes are expected to be online for ONLINECLIP. Correct the problem with the target volume(s).

CKZ44086I VALIDATING KEYWORD: keyword

Explanation: Parsing is checking the indicated keyword indicated in the command.

User response: No action is required.

CKZ45001I hh:mm:ss VARYOFF STARTED - PROGRAM REV=rrr | hh:mm:ss VARYOFF COMPLETED; RETURN CODE=nnn

Explanation: VARYOFF command processing message.

User response: No action is required.

CKZ45003I DDNAME=ddname ALLOCATED FOR DSN=datasetname

Explanation: 'ddname' has been dynamically allocated for the indicated data set.

User response: No action is required.

CKZ45004E DDNAME MISSING=ddname | OPEN FAILED FOR DDNAME=ddname

Explanation: 'ddname' was specified for DB2 Cloning Tool to use. Processing terminates.

User response: Either correct the ddname specified, or add the appropriate ddname to the job's JCL.

CKZ45005E ALLOCATION FAILED FOR DSN=datasetname

Explanation: Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ45005W DEALLOCATION FAILED FOR DDNAME=ddname

Explanation: Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ45006E ERROR CALLING CKZ01VV1 tttttt FUNCTION: function R15=nnnn R0=nnnnnnnn LOC=lllll

Explanation: A problem occurred using a dataspace. tttttt is the name of the internal table. llll is the

location where the error occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

CKZ45007W ERROR CALLING CKZ01HEX; FUNCTION: function R15=nnnn

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

CKZ45008E UNABLE TO LOAD PROGRAM= program name

Explanation: The indicated program name was not found. Processing terminates.

User response: Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

CKZ45009E ERROR ACCESSING JOURNAL FILE; LOC=lllll | ERROR ACCESSING VARY FILE; LOC=lllll

Explanation: A VSAM error occurred accessing the journal or vary file. Processing terminates.

User response: See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ45010E DUPLICATE VARY ENTRY; LOC=lllll

Explanation: A duplicate record was detected. Processing terminates.

User response: Verify that an empty vary file was used as input to the VARYOFF command. If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ45011E JOURNAL CONTROL RECORD NOT FOUND | JOURNAL VOLP RECORD NOT FOUND

Explanation: An expected record was not found in the DB2 Cloning Tool journal file. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ45012E JOURNAL CONTROL RECORD IS WRONG VERSION | JOURNAL VOLP RECORD IS WRONG VERSION

Explanation: The journal record does not match the expected format. The record is printed. Processing terminates.

User response: Verify that different releases of DB2 Cloning Tool have not been run using the same journal data set. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ45013E COUNT MISMATCH; RECORDS READ=rrrr CONTROL RECORD COUNT=cccc

Explanation: There was a problem with the journal records needed to initiate the command. The number of records read from the journal, rrrr, is not the same as the number indicated in the journal control record, cccc. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ45017E THE DDNAME IS EMPTY OR HAS BEEN DUMMIED, DDNAME: ddn

Explanation: No records were read from the ddname specified for a keyword. Processing terminates.

User response: Check that the DD has not been specified as 'DD DUMMY' or 'DD DSN=NULLFILE'. Check that the DSN specified in the ddn has been created successfully.

CKZ45018E THE DDNAME INPUT HAS EXCEEDED THE CURRENT CAPACITY, DDNAME: ddn

Explanation: The number of entries read from the ddname exceeded the current capacity. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ45019E DDNAME: ddn DOES NOT HAVE A LRECL OF 80

Explanation: The data set allocated to the ddname does not have a LRECL of 80. The LRECL of this data set must be 80. Processing terminates.

User response: Change the data set allocated to the ddname to have a LRECL of 80.

CKZ45020I COPY COMMAND: copy command

Explanation: Lists the cop command.

User response: No action is required.

CKZ45031E DEVICE NOT DEFINED; VOLSER: volser DEVICE NUMBER: dddd

Explanation: The device number that was specified is not defined to the z/OS system. Processing terminates.

User response: Change the specified device number to be a defined device defined to the z/OS system.

CKZ45032E VOLUME ALREADY OFFLINE; VOLSER: volser DEVICE NUMBER: dddd

Explanation: The device number that was specified is already offline to the z/OS system. Processing terminates if VOL-ALREADY-OFFLINE(QUIT) is specified.

User response: Bring the volume online or use VOL-ALREADY-OFFLINE(CONTINUE) if the volume should already be offline.

CKZ45033E WRONG VOLSER FOUND; FOUND VOLSER: volser EXPECTED VOLSER: volser DEVICE NUMBER: dddd

Explanation: The wrong volser was found on device dddd. Processing terminates.

User response: Correct the volume and device number specifications to match the current volser on the device.

CKZ45034E VOLUME NOT ONLINE AND DEVICE NUMBER IS NOT KNOWN; VOLSER: volser

Explanation: The volume is not online and a device number has not been specified for the volume. Processing terminates.

User response: Either bring the volume online or specify the device number of the volume if possible.

CKZ45035E DEVICE NOT DASD; VOLSER: volser DEVICE NUMBER: dddd

Explanation: The specified device is not defined to z/OS as a DASD device. Processing terminates.

User response: Change the device number to be that of a defined DASD device.

**CKZ45040E CKZ00900 UNEXPECTED RESULTS;
error text**

Explanation: An unexpected condition occurred calling program CKZ00900. 'error text' has a description of the problem. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZ45041E INVALID VALUE FOUND FOR item IN
keyword RECORD: value**

Explanation: An invalid value has been found for an item in a record in the data set allocated to the ddname for the keyword. The record is printed. Processing terminates.

User response: Correct the value for the item in the record to have a valid value.

**CKZ45048I OPTIONS IN EFFECT FOR THIS
EXECUTION: merged options**

Explanation: Informational message indicating how VARYOFF will handle the options. The displayed options are derived from the CKZINI and any overriding specifications in the command input.

User response: No action is required.

**CKZ45050E ERROR IN PARAMETERS FOR
keyword**

Explanation: The parameters for the indicated keyword were incorrect. Processing terminates.

User response: Check the keyword parameters. Mutually exclusive keywords may have been used.

**CKZ45051E REQUIRED KEYWORD MISSING:
keyword**

Explanation: A keyword required for processing has been omitted. Processing terminates.

User response: Specify the required keyword.

**CKZ45053E KEYWORD: keyword MAXIMUM
LENGTH: nnn EXCEEDED**

Explanation: The operand entered for a keyword exceeded the maximum length allowed for the operand. nnn is the maximum allowed length for the keyword. Processing terminates.

User response: Correct the length of the keyword's operand.

**CKZ45054E KEYWORD HAS MORE THAN 1
OPERAND; ONLY 1 ALLOWED:
keyword**

Explanation: Multiple operands were detected for a keyword; only one operand is permitted. Processing terminates.

User response: Correct the keyword to use one operand.

**CKZ45055E STORAGE GROUPS NOT SUPPORTED
WITH DATA-MOVER PROGRAM
NONE**

Explanation: The DATA-MOVER program was specified as 'NONE'. Storage group names/masks were specified for the source and/or target volume serials. Processing terminates.

User response: Correct the DATA-MOVER program specified, or, use keywords FROM-VOLSER/TO-VOLSER for the volume serials.

**CKZ45056E NOTHING SPECIFIED FOR
KEYWORD: keyword**

Explanation: A keyword was entered without an appropriate operand. Processing terminates.

User response: Specify an appropriate operand for the keyword.

**CKZ45057E DUPLICATE FOUND; KEYWORD:
keyword ENTRY: entry**

Explanation: The indicated 'entry' for the keyword was previously specified. Processing terminates.

User response: Remove the duplicate entry.

**CKZ45058E INVALID VALUE IN KEYWORD:
keyword VALUE: value error text**

Explanation: The value in the keyword is invalid. 'error text' indicates the problem detected with the value. Processing terminates.

User response: Correct the value specified in the keyword.

**CKZ45060E function ERROR; RETURN
CODE=nnnn REASON CODE=nnnn**

Explanation: An error occurred using the UCBLLOOK or UCBSKAN macro. Processing terminates.

User response: A return code 4 from UCBLLOOK may indicate the volser is offline. If unable to determine the cause of the error, contact IBM Software Support. Have available the listing containing this message.

**CKZ45061E CKZ01SMF ERROR; RETURN
CODE=nnnn LOC: llllll entry**

Explanation: An error occurred using CKZ01SMF to obtain SSI information for the 'entry'. llllll is the internal location where the error occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing containing this message.

**CKZ45062E NO STORAGE GROUPS RETURNED
BY SSI**

Explanation: The VARYOFF command was requested to determine volume serial numbers by storage group names. SSI did not return any storage groups to CKZ01SMF. Processing terminates.

User response: Check that storage groups are defined on the system. If unable to determine the cause of this message, contact IBM Software Support. Have available the listing containing this message.

**CKZ45063W NO STORAGE GROUP MATCH
FOUND; ENTRY: entry | NO VOLUME
MATCH FOUND; ENTRY: entry**

Explanation: The indicated 'entry' for the keyword was not matched. For STORAGE GROUP, no matches were found for the mask. For VOLUME, the indicated volser, or, the volser derived from a storage group, was not found. Processing continues.

User response: No action is required.

**CKZ45063E EXPLICIT STORAGE GROUP NOT
FOUND; ENTRY: entry**

Explanation: The indicated storage group name is not defined as a storage group. Processing terminates.

User response: Correct to specify defined storage group name.

**CKZ45067W NO VOLUME SERIALS FOR
STORAGE GROUP: storage group**

Explanation: SSI did not return any volsers for the storage group to CKZ01SMF. Processing continues.

User response: No action is required.

**CKZ45070E THE COPY COMMAND HAS
UNBALANCED PARENTHESES**

Explanation: The copy command has unbalanced parentheses. Processing terminates.

User response: Correct the copy command to have balanced parentheses.

**CKZ45071E THE COPY COMMAND HAS A
KEYWORD THAT IS TOO LONG:
keyword**

Explanation: The copy command has a keyword specified that is longer than a keyword is expected to be. Processing terminates.

User response: Correct the copy command keyword to be valid.

**CKZ45072E THE COPY COMMAND IS NOT A
COPY COMMAND**

Explanation: The copy command does not appear to be a copy command. Processing terminates.

User response: Correct the copy command.

CKZ45073I PROCESSING COPY COMMAND

Explanation: The copy command is being read and parsed.

User response: No action is required.

**CKZ45074E COPY COMMAND HAS MUTUALLY
EXCLUSIVE KEYWORDS: keyword1
keyword2**

Explanation: The copy command has keywords specified that are mutually exclusive. Processing terminates.

User response: Correct the copy command to not have mutually exclusive keywords specified.

**CKZ45075E UNMATCHED ENTRIES IN COPY
COMMAND KEYWORD: keyword**

Explanation: For VOLPAIRS, there must be a source volume serial, target volume serial. An uneven number of entries was specified. For VOLPAIRSDEVN, there must be a source volume serial, target volume serial, target device number. Unmatched entries were found. Processing terminates.

User response: Correct the keyword specification.

CKZ45076E NO VOLUMES SERIALS SELECTED

Explanation: No volumes were selected for processing. Processing terminates.

User response: Correct the copy command so volumes will be selected for processing.

**CKZ45077I VOLUMES SELECTED FOR VARY
OFFLINE PROCESSING: volser dddd**

Explanation: Lists the volser and device number of the volumes selected for processing.

User response: No action is required.

CKZ45078E VOLUME PAIRS ONLY ACCEPTED WITH DATA-MOVER PROGRAM NONE

Explanation: Keyword VOLPAIRS, VOLPAIRS-DDN, VOLPAIRSDEVN, VOLPAIRSDEVN-DDN, VOLPAIRSDEVN-NOCLIP, or VOLPAIRSDEVN-NOCLIP-DDN was used, but, the DATA-MOVER program is not 'NONE'. Processing terminates.

User response: Correct the keyword specifications.

CKZ45086I VALIDATING KEYWORD: keyword

Explanation: Parsing is checking the indicated keyword indicated in the command.

User response: No action is required.

CKZ45101I hh:mm:ss VARY VOLUMES STARTED - PROGRAM REV=rrr | hh:mm:ss VARY VOLUMES COMPLETED; RETURN CODE=nnn

Explanation: VARYON command processing message.

User response: No action is required.

CKZ45106E ERROR CALLING CKZ01VV1 tttttt FUNCTION: function R15=nnnn R0=nnnnnnnn LOC=lllll

Explanation: A problem occurred using a dataspace. tttttt is the name of the internal table. lllll is the location where the error occurred. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message and the CKZINI member.

CKZ45107W ERROR CALLING CKZ01HEX; FUNCTION: function R15=nnnn

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

CKZ45120I CONSOLE name ACQUIRED FOR SYSPLEX VARY COMMANDS

Explanation: A console session has been acquired so that sysplex vary commands can be issued.

User response: No action is required.

CKZ45121I CONSOLE name FREED

Explanation: The operator console has been freed.

User response: No action is required.

CKZ45122W UNABLE TO ACQUIRE A CONSOLE

Explanation: DB2 Cloning Tool has failed to acquire a console for performing operator commands. Processing continues without using a console to perform operator commands.

User response: No action is required.

CKZ45123I CONSOLE name ALREADY IN USE, WILL TRY ANOTHER

Explanation: The displayed name is already in use, probably from another copy of DB2 Cloning Tool. DB2 Cloning Tool will increment the number portion of the name and try again.

User response: No action is required.

CKZ45124W MCS ALERT RECEIVED; text

Explanation: An alert has been received for the console. Text describes the type of alert. Processing continues.

User response: Please report this message to IBM Software Support.

CKZ45125I COMMAND: text

Explanation: Display operator command being issued.

User response: No action is required.

CKZ45126W WAIT TIME EXCEEDED FOR COMMAND RESPONSE

Explanation: A response to the operator command was not received in a timely manner. Processing continues.

User response: Please report this message to IBM Software Support.

CKZ45127I NON RESPONSE MDBS RECEIVED: nnn

Explanation: Informational message that displays the number of received messages that were not a response to the command issued.

User response: No action is required.

CKZ45131W MCSOPER ERROR; FUNCTION=
function RC=nnnn RSN=nnnn |
MCSOPMSG ERROR; FUNCTION=
function RC=nnnn RSN=nnnn

Explanation: An error occurred using the MCSOPER or MCSOPMSG macro. Processing continues.

User response: Contact IBM Software Support. Have available the listing containing this message.

CKZ45136E IEEVARYD VARY OFFLINE FAILED
FOR VOLSER: volser DEVICE: dddd;
INVALID PARAMETERS | IEEVARYD
VARY ONLINE FAILED FOR VOLSER:
volser DEVICE: dddd; INVALID
PARAMETERS

Explanation: The parameters given to IEEVARYD are incorrect. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ45137E IEEVARYD VARY OFFLINE FAILED
FOR VOLSER: volser DEVICE: dddd
RETURN CODE: nnnnnnnn REASON
CODE: nnnnnnnn | IEEVARYD VARY
ONLINE FAILED FOR VOLSER: volser
DEVICE: dddd RETURN CODE:
nnnnnnnn REASON CODE: nnnnnnnn

Explanation: The vary for the indicated device failed. Processing terminates.

User response: Correct the problem causing the vary to fail. If unable to correct the problem, Contact IBM Software Support. Have available the listing that contains this message.

CKZ45138E IEEVARYD VARY OFFLINE FAILED
FOR VOLSER: volser DEVICE: dddd
R15: nnnnnnnn | IEEVARYD VARY
ONLINE FAILED FOR VOLSER: volser
DEVICE: dddd R15: nnnnnnnn

Explanation: The vary for the indicated device failed. Processing terminates.

User response: Correct the problem causing the vary to fail. If unable to correct the problem, Contact IBM Software Support. Have available the listing that contains this message.

CKZ45140I NO VOLUMES NEEDED LOCAL VARY
OFFLINE | NO VOLUMES NEEDED
LOCAL VARY ONLINE

Explanation: No volumes needed to be varied on the local system.

User response: No action is required.

CKZ45141I VOLUME SERIAL: volser DEVICE
NUMBER: dddd IS NOW OFFLINE |
VOLUME SERIAL: volser DEVICE
NUMBER: dddd IS NOW ONLINE

Explanation: The indicated device is now offline or online to the current system.

User response: No action is required.

CKZ45144E DEVICE NUMBER: dddd FOR TARGET
VOLUME: volser DOES NOT EXIST

Explanation: The indicated device number is not defined to the z/OS system. Processing terminates.

User response: Correct the device number to use a defined device.

CKZ45145E DEVICE NUMBER: dddd IS ONLINE
WITH UNEXPECTED VOLUME
SERIAL: volser

Explanation: The indicated device number is online but the volser of the device does not match what is expected. Processing terminates.

User response: Either correct the device number to use the correct device or correct the device to have the correct contents.

CKZ45146W DEVICE NUMBER: dddd FOR
VOLUME: volser IS STILL OFFLINE

Explanation: The vary online got a good return code but the device is still offline. Processing continues.

User response: Determine why the volume went offline after it was varied online. If unable to determine the cause, contact IBM Software Support. Have available the listing containing this message and the system log from the time this happened.

CKZ45147I VOLUME SERIAL: volser DEVICE
NUMBER: dddd WILL BE VARIED
OFFLINE | VOLUME SERIAL: volser
DEVICE NUMBER: dddd WILL BE
VARIED ONLINE

Explanation: Display the vary that will be done.

User response: No action is required.

CKZ45160E ERROR DURING UCBLLOOK FOR
DEVICE NUMBER dddd VOLUME
SERIAL: volser RETURN CODE=nnnn
REASON CODE=nnnn

Explanation: An error occurred using the UCBLLOOK macro. Processing

User response: A return code 4 from UCBLLOOK may indicate the volser is offline. If unable to determine the

cause of the error, contact IBM Software Support. Have available the listing containing this message.

CKZ46001I hh:mm:ss VARYON STARTED - PROGRAM REV=rrr | hh:mm:ss VARYON COMPLETED; RETURN CODE=nnn

Explanation: VARYON command processing message.

User response: No action is required.

CKZ46003I DDNAME=ddname ALLOCATED FOR DSN=datasetname

Explanation: 'ddname' has been dynamically allocated for the indicated data set.

User response: No action is required.

CKZ46004E DDNAME MISSING=ddname

Explanation: 'ddname' was specified for DB2 Cloning Tool to use. Processing terminates.

User response: Either correct the ddname specified, or add the appropriate ddname to the job's JCL.

CKZ46005E ALLOCATION FAILED FOR DSN: datasetname

Explanation: Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ46005W DEALLOCATION FAILED FOR DDNAME: ddname

Explanation: Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ46007W ERROR CALLING CKZ01HEX; FUNCTION: function R15=nnnn

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

CKZ46008E UNABLE TO LOAD PROGRAM: program name

Explanation: The indicated program name was not found. Processing terminates.

User response: Check that the job's //STEPLIB library is correct. If unable to resolve the problem, contact IBM Software Support.

CKZ46009E ERROR ACCESSING JOURNAL FILE; LOC=IIIII | ERROR ACCESSING VARY FILE; LOC=IIIII

Explanation: A VSAM error occurred accessing the journal or vary file. Processing terminates.

User response: See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ46011E JOURNAL CONTROL RECORD NOT FOUND | JOURNAL VOLP RECORD NOT FOUND | VARY CONTROL RECORD NOT FOUND | VARY VOL RECORD NOT FOUND

Explanation: An expected record was not found in the DB2 Cloning Tool journal file or the DB2 Cloning Tool vary file. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZ46012E JOURNAL CONTROL RECORD IS WRONG VERSION | JOURNAL VOLP RECORD IS WRONG VERSION | VARY CONTROL RECORD IS WRONG VERSION | VARY VOL RECORD IS WRONG VERSION

Explanation: The journal or vary record does not match the expected format. The record is printed. Processing terminates.

User response: Verify that different releases of DB2 Cloning Tool have not been run using the same journal or vary data set. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains this message.

CKZ46013E COUNT MISMATCH; RECORDS READ=rrrr CONTROL RECORD COUNT=cccc

Explanation: There was a problem with the journal or vary records needed to initiate the command. The number of records read from the journal or vary, rrrr, is not the same as the number indicated in the journal or vary control record, cccc. Processing terminates.

User response: Contact IBM Software Support. Have

available the listing that contains this message.

**CKZ46031E DEVICE NOT DEFINED; VOLSER:
volser DEVICE NUMBER: dddd**

Explanation: The device number that was specified is not defined to the z/OS system. Processing terminates.

User response: Change the specified device number to be a defined device defined to the z/OS system.

**CKZ46032E VOLUME ALREADY ONLINE;
VOLSER: volser DEVICE NUMBER:
dddd**

Explanation: The device number that was specified is already online to the z/OS system. Processing terminates if VOL-ALREADY-ONLINE(QUIT) is specified.

User response: Bring the volume offline or use VOL-ALREADY-ONLINE(CONTINUE) if the volume should already be online.

**CKZ46033E WRONG VOLSER FOUND; FOUND
VOLSER: volser EXPECTED VOLSER:
volser DEVICE NUMBER: dddd**

Explanation: The wrong volser was found on device devn. Processing terminates.

User response: Correct the volume and device number specifications to match the current volser on the device.

**CKZ46034E DEVICE NUMBER IS NOT KNOWN;
VOLSER: volser**

Explanation: The volume is not online and a device number has not been specified for the volume. The device number is needed to vary the volume online. Processing terminates.

User response: Either bring the volume online or specify the device number of the volume if possible.

**CKZ46048I OPTIONS IN EFFECT FOR THIS
EXECUTION: merged options**

Explanation: Informational message indicating how VARYON will handle the options. The displayed options are derived from the CKZINI and any overriding specifications in the command input.

User response: No action is required.

**CKZ46050E ERROR IN PARAMETERS FOR
keyword**

Explanation: The parameters for the indicated keyword were incorrect. Processing terminates.

User response: Check the keyword parameters. Mutually exclusive keywords may have been used.

**CKZ46051E REQUIRED KEYWORD MISSING:
keyword**

Explanation: A keyword required for processing has been omitted. Processing terminates.

User response: Specify the required keyword.

**CKZ46053E KEYWORD: keyword MAXIMUM
LENGTH: nnn EXCEEDED**

Explanation: The operand entered for a keyword exceeded the maximum length allowed for the operand. nnn is the maximum allowed length for the keyword. Processing terminates.

User response: Correct the length of the keyword's operand.

**CKZ46054E KEYWORD HAS MORE THAN 1
OPERAND; ONLY 1 ALLOWED:
keyword**

Explanation: Multiple operands were detected for a keyword; only one operand is permitted. Processing terminates.

User response: Correct the keyword to use one operand.

**CKZ46056E NOTHING SPECIFIED FOR
KEYWORD: keyword**

Explanation: A keyword was entered without an appropriate operand. Processing terminates.

User response: Specify an appropriate operand for the keyword.

**CKZ46058E INVALID VALUE IN KEYWORD:
keyword VALUE: value error text**

Explanation: The value in the keyword is invalid. 'error text' indicates the problem detected with the value. Processing terminates.

User response: Correct the value specified in the keyword.

**CKZ46060E function ERROR; RETURN
CODE=nnnn REASON CODE=nnnn**

Explanation: An error occurred using the UCBLOOK macro. Processing terminates.

User response: A return code 4 from UCBLOOK may indicate the volser is offline. If unable to determine the cause of the error, contact IBM Software Support. Have available the listing containing this message.

CKZ46077I VOLUMES SELECTED FOR VARY
ONLINE PROCESSING: volser dddd

Explanation: Lists the volser and device number of the volumes selected for processing.

User response: No action is required.

CKZ46086I VALIDATING KEYWORD: keyword

Explanation: Parsing is checking the indicated keyword indicated in the command.

User response: No action is required.

CKZ48001I hh:mm:ss JRNLUPGRADE STARTED -
PROGRAM REV=rrr | hh:mm:ss
JRNLUPGRADE COMPLETED;
RETURN CODE=nnn

Explanation: JRNLUPGRADE command processing message.

User response: No action is required.

CKZ48003I DDNAME=ddname ALLOCATED FOR
DSN=datasetname

Explanation: 'ddname' has been dynamically allocated for the indicated data set.

User response: No action is required.

CKZ48004E DDNAME MISSING=ddname

Explanation: 'ddname' was specified for DB2 Cloning Tool to use. Processing terminates.

User response: Either correct the ddname specified, or add the appropriate ddname to the job's JCL.

CKZ48005E ALLOCATION FAILED FOR DSN:
datasetname

Explanation: Dynamic allocation for a data set failed. The associated z/OS messages are displayed. Processing terminates.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ48005W DEALLOCATION FAILED FOR
DDNAME: ddname

Explanation: Dynamic deallocation for a ddname failed. The associated z/OS messages are displayed. Processing continues.

User response: If unable to determine the reason for the failure from the associated z/OS messages, contact IBM Software Support. Have available the listing containing these messages.

CKZ48007W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn

Explanation: An error occurred using CKZ01HEX to print a record. Processing continues.

User response: Please report this message to IBM Software Support.

CKZ48009E ERROR ACCESSING JOURNAL FILE;
DDN=ddname LOC=lllll

Explanation: A VSAM error occurred accessing a journal file. Processing terminates.

User response: See associated CKZVSEnnE error messages. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ48010E DUPLICATE JOURNAL ENTRY;
DDN=ddname LOC=lllll

Explanation: A duplicate record was detected in the NEW journal file. Processing terminates.

User response: Verify the referenced NEW journal file was empty when the JRNLUPGRADE command started processing. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ48020W UNKNOWN JOURNAL RECORD
TYPE: type

Explanation: The OLD journal contains an unknown record type. The record is copied to the NEW journal. Not all journal records need to be upgraded for the scenarios documented where JRNLUPGRADE can be used.

User response: Verify the current version of the JRNLUPGRADE command is being used. Please report this message to IBM Software Support.

CKZ48021E UNKNOWN JOURNAL RECORD
VERSION

Explanation: The OLD journal contains an unknown version of a record. The record is printed. Processing terminates.

User response: Verify the current version of the JRNLUPGRADE command is being used. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains these messages.

CKZ48031I nnn TOTAL RECORDS READ

Explanation: Shows the total number of records read from the OLD journal.

User response: No action is required.

CKZ48032I nnn TOTAL RECORDS WRITTEN

Explanation: Shows the total number of records written to the NEW journal.

User response: No action is required.

CKZ48033I nnn type RECORDS UPGRADED

Explanation: Shows the total number of records of the displayed type that have been upgraded to the current level.

User response: No action is required.

CKZ48034I nnn type RECORDS ALREADY AT CURRENT LEVEL

Explanation: Shows the total number of records of the displayed type that did not need to be upgraded because they were already at the current level.

User response: No action is required.

CKZ48035I nnn UNKNOWN RECORDS

Explanation: Shows the total number of records of unknown type that were copied without change to the NEW journal.

User response: No action is required.

CKZ48051E REQUIRED KEYWORD MISSING: keyword

Explanation: A keyword required for processing has been omitted. Processing terminates.

User response: Specify the required keyword.

CKZ48053E KEYWORD: keyword MAXIMUM LENGTH: nnn EXCEEDED

Explanation: The operand entered for a keyword exceeded the maximum length allowed for the operand. nnn is the maximum allowed length for the keyword. Processing terminates.

User response: Correct the length of the keyword's operand.

CKZ48054E KEYWORD HAS MORE THAN 1 OPERAND; ONLY 1 ALLOWED: keyword

Explanation: Multiple operands were detected for a keyword; only one operand is permitted. Processing terminates.

User response: Correct the keyword to use one operand.

CKZ48056E NOTHING SPECIFIED FOR KEYWORD: keyword

Explanation: A keyword was entered without an appropriate operand. Processing terminates.

User response: Specify an appropriate operand for the keyword.

CKZ48058E INVALID VALUE IN KEYWORD: keyword VALUE: value

Explanation: The value in the keyword is invalid. 'error text' indicates the problem detected with the value. Processing terminates.

User response: Correct the value specified in the keyword.

CKZ50001E PRODUCT INIT FAILURE, RC=rrr, RS=sss

Explanation: Initialization has failed. rrr = return code sss = reason code

User response: Find one or more detailed warning or failure messages. Use the more detailed message(s) to determine the cause.

| **CKZ50020I The number of subtasks was changed to 1 for the source job as PGM(SRCIMCPY)**

| **Explanation:** When cloning from image copies (PGM(SRCIMCPY), a single subtask is required.

| **User response:** No action is required.

CKZ50003I TCP SERVER JOB, NUMBER OF SUBTASKS RESET TO 1

Explanation: MAX_SUBTASKS in PARMLIB set to a value greater than 1 when starting a TCPIP server job. The value has been changed to a 1.

User response: No action is required.

CKZ50004E DISCOVERY PHASE HAS FAILED

Explanation: One or more errors has occurred during the source job discovery phase. The job terminates.

User response: Find one or more detailed warning or failure messages. Use the more detailed message(s) to determine the cause.

CKZ50005W DISCOVERY PHASE ENDED WITH WARNING(S)

Explanation: One or more warning messages were issued during the source job discovery phase.

User response: Find one or more detailed warning messages. Use the more detailed message(s) to

CKZ50006E • CKZ50092E

determine the cause and correct if needed.

CKZ50006E SUBTASK nn, IS HUNG, UNABLE TO EXIT

Explanation: The hang may or may not be an internal problem. nn = subtask number

User response: Call IBM Software Support unless the hang is the result of an MVS or DB2 problem.

CKZ50007E UNEXPECTED ATTACH ERROR, RC=rrr, RS=sss rrr = return code sss = reason code

Explanation: This is an internal error.

User response: Call IBM Software Support.

CKZ50008E UNABLE TO ALLOCATE A TDE

Explanation: This is an internal error.

User response: Call IBM Software Support.

CKZ50009E UNABLE TO ATTACH SUBTASK

Explanation: This is probably an internal error.

User response: Call IBM Software Support.

CKZ50010E QUEUE SHOW ERROR, RS=sss

Explanation: This is an internal error. sss = reason code

User response: Call IBM Software Support.

CKZ50011I PROCESSING COMPLETE WITH NO ERRORS

Explanation: This message is informational.

User response: No action is required.

CKZ50012E COMPLETED, WITH ERRORS, RC=rrr, RS=sss

Explanation: CKZ completed but with errors. rrr = return code sss = reason code

User response: Find one or more detailed failure messages. Use the more detailed message(s) to determine the cause.

CKZ50013E Completed, with Warnings, RC=rrr, RS=sss

Explanation: CKZ completed but with warnings. rrr = return code sss = reason code

User response: Find one or more detailed warning messages. Use the more detailed message(s) to determine the cause.

CKZ50016I mmm ddd ttt VERS=vvv nnn...nnn

Explanation: Module level. mmm = LMOD name ddd = assembly date ttt = assembly time vvv = version nnn = fields unused

User response: No action is required.

CKZ50017I The number of subtasks changed to 1 for the source job as Enable-Catalog-Prefetch is Y.

Explanation: Note that the target job will run with the number that was specified with the parm.

User response: No action is required.

CKZ50018E When SUBTASK-DATASET-EXTENSIONS is Y, *feature_name* cannot be enabled.

Explanation: The functionality listed in the message is not compatible with the SUBTASK-DATASET-EXTENSIONS(Y) parameter and cannot be enabled.

User response: Disable the feature listed in the message and resubmit the job.

CKZ50019I STOP command accepted, job *started_task_name*, *jobID* *job_id_for_started_task*.

Explanation: The started task specified with the job name parm is processing the requested STOP command.

User response: No action is required.

CKZ50090I INITIALIZATION COMPLETE: sss

Explanation: This is an informational WTO. sss = EXEC module name, version and release, assembly date and time

User response: No action is required.

CKZ50091E INITIALIZATION FAILED

Explanation: This is an error WTO.

User response: Check detailed error message(s) for the reason. Check the ACTION field of the detailed message(s).

CKZ50092E ERROR TAKING SVC DUMP: rrr

Explanation: This is an error WTO. rrr = R15 return from SDUMP

User response: Call IBM Software Support if unable to resolve this error.

CKZ50093E ERROR TAKING SVC DUMP: rrr

Explanation: This is an error WTO. This SVC was attempted from the ESTAE routine rrr = R15 return from SDUMP

User response: Call IBM Software Support if unable to resolve this error.

CKZ50094E ERROR TAKING SVC DUMP: rrr

Explanation: This is an error WTO. This SVC was attempted from the ESTAE REPLY routine. rrr = R15 return from SDUMP

User response: Call IBM Software Support if unable to resolve this error.

CKZ50095I NO ESTAE RETRY ALLOWED PER SDWA

Explanation: This is an informational WTO.

User response: No action is required.

CKZ50098E *message text*

Explanation: This is an error WTO from initialization. It is followed by CKZ50099E.

message text = error text:

- allocate BVT failed
- allocate save areas failed
- parameter parsing error
- parameter value error, see CKZPRINT
- length and/or count parm exceeds 32k-1
- error return from INITIALIZATION
- ESTAE non zero RC

User response: Call IBM Software Support if unable to resolve this error.

CKZ50099E INITIALIZATION FAILED

Explanation: This is an error WTO. It follows WTO CKZ50098E.

User response: Call IBM Software Support if unable to resolve this error.

CKZ50601I DSS LEVEL IS hhh, ttt SUPPORT FASTREPLICATE

Explanation: This message is informational.

hhh = DSS level

ttt = Does or Does Not

User response: If the text is Does Not, then DSS will initiate normal copies without using FASTREPLICATE.

CKZ50602E ANTMAIN IS NOT ACTIVE

Explanation: This is a system environment error. ANTMAIN is the System Data Mover.

User response: Contact your system programmer.

CKZ50603E ANTRQST ERROR: ttt RC=rrr, RS=sss, VER=lll

Explanation: This is a system environment error. ANTMAIN is the System Data Mover.

- ttt = VERSION of DATA-MOVER
- rrr = return code
- sss = reason code
- lll = the level of ANTMAIN

User response: Contact your system programmer.

CKZ50604W ANTRQST LEVEL III IS NOT SUPPORTED

Explanation: This is a system environment error. ANTMAIN is the System Data Mover. The level returned was unexpected. lll = the level of ANTMAIN

User response: Contact your system programmer.

CKZ50605I ANTRQST LEVEL IS III

Explanation: This message is informational. lll = the level of ANTMAIN

User response: No action is required.

CKZ52701I LOGAPPLY Command *command_name* is *string_value*

Explanation: This message displays the string value of the target input log apply command.

User response: No action is required.

CKZ52702I LOGAPPLY Command *command_name* is *hex_value*

Explanation: This message displays the hex value of the target input log apply command.

User response: No action is required.

CKZ52703I LOGAPPLY Command *command_name* is *decimal_value*

Explanation: This message displays the decimal value of the target input log apply command.

User response: No action is required.

CKZ52704E • CKZ53083E

CKZ52704E LOGAPPLY Command Parsing Error,
RC=*return_code*, RS=*reason_code*

Explanation: The LOGAPPLY target input command cannot be parsed.

User response: Correct the error and resubmit the job.

CKZ52705E LOGAPPLY *command_name*,
command_value Invalid Length

Explanation: The LOGAPPLY target input command has an invalid length.

User response: Correct the error and resubmit the job.

CKZ52706E MINILOG-HLQ has an Invalid Value,
value

Explanation: The LOGAPPLY MINILOG-HLQ has an invalid value.

User response: Correct the error and resubmit the job.

CKZ52707I LOGAPPLY Command Data Sharing
Member, ID=*data_sharing_ID*
SSID=*source_subsystem_for_this_ID*,
ZPARAM=*ZPARAM_member_for_this_ID*,
BSDS01=*BSDS01_data_set_for_this_ID*,
BSDS02=*BSDS02_data_set_for_this_ID*

Explanation: The data sharing member information.

User response: No action is required.

CKZ52708E *parameter_string* has an Invalid Value,
value

Explanation: The LOGAPPLY parm has an invalid value.

User response: Correct the error and resubmit the job.

CKZ53001I BEGIN COMMAND FILE SYNTAX
CHECKING

Explanation: This message is informational.

User response: No action is required.

CKZ53002E SI09 ERROR RETURN, rrr

Explanation: This may or may not be a user error. CKZIN can not be read. rrr = return code

User response: Call IBM Software Support if unable to resolve this error.

CKZ53006E CKZIN COMMAND *ccc*, NO
OPERAND

Explanation: This is a user error. *ccc* = command

User response: Correct the input and resubmit the job.

CKZ53007E NO COMMANDS FOUND

Explanation: This is a user error.

User response: Correct the input and resubmit the job.

CKZ53008I END COMMAND FILE SYNTAX
CHECKING

Explanation: This message is informational.

User response: No action is required.

CKZ53009E PGM(SRCIMCPY) was specified,
command *command* disabled

Explanation: An invalid command was specified with PGM(SRCIMCOPY). The following commands are not valid with PGM(SRCIMCOPY): SIM(A), DATA-MASKING, SUBTASK-DATASET-EXTENSIONS and USE-RUNTIME-REPOSITORY.

User response: Change the invalid command to the default and resubmit the job.

CKZ53080E mmm SUPPORT MODULE MISSING

Explanation: This is an error WTO. mmm = module name

User response: Ensure the product is installed correctly and STEPLIB points to the correct CKZ libraries.

CKZ53081E dd DD STATEMENT MISSING

Explanation: This is an error WTO. dd = the missing DD card

User response: Add the DD to the JCL and resubmit the job.

CKZ53082E FATAL ERRORS HAVE OCCURRED
DURING //CKZINI PROCESSING

Explanation: This is a probable user error that has occurred with the initial processing of Parmlib.

User response: Correct the Parmlib input and resubmit the job.

CKZ53083E MUST EXECUTE AS A Z/OS
AUTHORIZED PROGRAM

Explanation: This is a user error.

User response: Authorize the CKZLOADLIB and resubmit the job.

CKZ53084E UNABLE TO LOAD PROGRAM: ppp

Explanation: This is a user error. ppp = module unable to load

User response: After successful installation of CKZ, resubmit the job.

CKZ53091E FATAL ERRORS HAVE OCCURRED DURING //CKZINI PROCESSING

Explanation: This is probably a user error.

User response: Correct the error in Parmlib and resubmit the job.

CKZ53092W DSS_COPY_COMMANDS PARM HAS AN INVALID VALUE, ddd, DEFAULTING TO EEE

Explanation: This is a user error. ddd = decimal value eee = default decimal value

User response: Correct the input and resubmit the job or use the default value (24).

CKZ53093W DSS_COPY_COMMANDS PARM HAS AN INVALID VALUE, vvv, DEFAULTING TO eee

Explanation: This is a user error. vvv = non decimal value eee = default decimal value

User response: Correct the input and resubmit the job or use the default value.

CKZ53094I DSS_COPY_COMMANDS PARM NOT IN PARMLIB, DEFAULTING TO eee

Explanation: This message is informational. eee = default decimal value

User response: No action is required.

CKZ53095I PARMLIB DSS_COPY_COMMANDS SET TO ddd

Explanation: This message is informational. ddd = decimal value

User response: No action is required.

CKZ53096W DSNS_PER_COPY PARM HAS AN INVALID VALUE, ddd, DEFAULTING TO eee

Explanation: This is a user error. ddd = decimal value eee = default value

User response: Correct the input and resubmit the job or use the default value.

CKZ53097W DSNS_PER_COPY PARM HAS AN INVALID VALUE, vvv, DEFAULTING TO eee

Explanation: This is a user error. vvv = non decimal value eee = default value

User response: Correct the input and resubmit the job or use the default value.

CKZ53098I DSNS_PER_COPY PARM NOT IN PARMLIB, DEFAULTING TO eee

Explanation: This message is informational. eee = default decimal value

User response: No action is required.

CKZ53099I PARMLIB DSNS_PER_COPY SET TO ddd

Explanation: This message is informational. ddd = decimal value

User response: No action is required.

CKZ53201I MAX_SUBTASKS PARM NOT IN PARMLIB, DEFAULTING TO 1

Explanation: This message is informational.

User response: No action is required.

CKZ53202E DD NAME ddname IN fff NOT FOUND IN JCL

Explanation: This is a user error. ddname = DD name not found in the JCL fff = COPY parm with the bad DD name

User response: Correct the JCL or the COPY Command DD Name and resubmit the source job.

CKZ53203W MAX_SUBTASKS PARM HAS AN INVALID VALUE, vvv, DEFAULTING TO 1

Explanation: This is a user error. vvv = bad subtask value

User response: Correct the MAX_SUBTASK value in Parmlib and resubmit the source job.

CKZ53204W MAX_SUBTASKS PARM HAS AN INVALID VALUE, ddd, DEFAULTING TO 1

Explanation: This is a user error. ddd = decimal value

User response: Correct the input and resubmit the job or use the default value.

CKZ53205I PARMLIB MAX_SUBTASKS SET TO ddd

Explanation: This message is informational. ddd = decimal value

User response: No action is required.

CKZ53206I TCPIP_SERVER_PORT PARM NOT IN PARMLIB, DEFAULTING TO 5099

Explanation: This message is informational.

User response: No action is required.

CKZ53207W TCPIP_SERVER_PORT PARM HAS AN INVALID VALUE ddd, DEFAULTING TO 5099

Explanation: This is a user error. ddd = decimal value

User response: Correct the input and resubmit the job or use the default value.

CKZ53210I PARMLIB TCPIP_SERVER_PORT SET TO ddd

Explanation: This message is informational. ddd = decimal value

User response: No action is required.

CKZ53211I MAX_RC PARM NOT IN PARMLIB, DEFAULTING TO 0

Explanation: This message is informational.

User response: No action is required.

CKZ53212W MAX_RC PARM HAS AN INVALID VALUE, ddd, DEFAULTING TO 0

Explanation: This is a user error. ddd = decimal value

User response: Correct the input and resubmit the job or use the default value.

CKZ53214I PARMLIB MAX_RC SET TO ddd

Explanation: This message is informational. ddd = decimal value

User response: No action is required.

CKZ53215I MAX_COPY_RC PARM NOT IN PARMLIB, DEFAULTING TO 0

Explanation: This message is informational.

User response: No action is required.

CKZ53216W MAX_COPY_RC PARM HAS AN INVALID VALUE, ddd, DEFAULTING TO 0

Explanation: This is a user error. ddd = decimal value

User response: Correct the input and resubmit the job or use the default value.

CKZ53218I PARMLIB MAX_COPY_RC SET TO ddd

Explanation: This message is informational. ddd = decimal value

User response: No action is required.

CKZ53219I DB2_COMMAND_RESPONSE_WAIT PARM NOT IN PARMLIB, DEFAULTING TO ddd SECONDS

Explanation: This message is informational. ddd = decimal value

User response: No action is required.

CKZ53220W DB2_COMMAND_RESPONSE_WAIT Parm has an Invalid Value, ddd, Defaulting to ddd Seconds

Explanation: This message is a warning. ddd = decimal value

User response: Correct the input and resubmit the job or use the default value.

CKZ53223I Parmlib DB2_COMMAND_RESPONSE_WAIT set to ddd Seconds

Explanation: This message is informational. ddd = decimal value

User response: No action is required.

CKZ53224I DB2_PLAN Parm Not in Parmlib, Defaulting to CKZPLAN

Explanation: This message is informational.

User response: No action is required.

CKZ53225W DB2_PLAN Parm has an Invalid Value, vvv, Defaulting to CKZPLAN

Explanation: This is a user error. vvv = DB2 CKZ plan name

User response: Correct the input and resubmit the job or use the default value.

CKZ53226I Parmlib DB2_PLAN set to vvv

Explanation: This message is informational. vvv = DB2 CKZ plan name

User response: No action is required.

CKZ53227I COPY_OPTION ccc Not in Parmlib, Defaulting to sss

Explanation: This message is informational. ccc = COPY_OPTION string sss = Y or N default value

User response: No action is required.

CKZ53228W COPY_OPTION ccc has an Invalid Value, vvv, Defaulting to N

Explanation: This is a user error. ccc = COPY_OPTION string vvv = COPY_OPTION value

User response: Correct the input and resubmit the job or use the default value.

CKZ53229I Parmlib COPY_OPTION ccc set to vvv

Explanation: This message is informational. ccc = COPY_OPTION string vvv = COPY_OPTION value

User response: No action is required.

CKZ53231I TCPIP_STC_NAME Parm Not in Parmlib, Defaulting to TCPIP

Explanation: This message is informational.

User response: No action is required.

CKZ53232W TCPIP_STC_NAME Parm has an Invalid Value, sss, Defaulting to TCPIP

Explanation: This is a user error.

User response: Correct the input and resubmit the job or use TCPIP as the default.

CKZ53235I Parmlib TCPIP_STC_NAME set to sss

Explanation: This message is informational. sss = name of TCPIP started task

User response: No action is required.

CKZ53501E DATA-MOVER PROGRAM ppp IS INVALID

Explanation: This is a user error. ppp = program name

User response: Correct the input and resubmit the job.

CKZ53502I COPY TARGET, SSID sss, LOCATION lll, USERID uuu, PASSWORD ppp

Explanation: This message is informational.

- sss = target DB2 subsystem
- lll = target DB2 subsystem location
- uuu = user id for DDF
- ppp = password for DDF (asterisks)

User response: No action is required.

CKZ53503I COPY DEFAULT VCAT IS sss

Explanation: This message is informational. sss = VCAT

User response: No action is required.

CKZ53504I SERVER IPV4 IS IPADDR

Explanation: This message is informational.

User response: No action is required.

CKZ53505I SERVER PORT IS DDD

Explanation: This message is informational. ddd = decimal number

User response: No action is required.

CKZ53506I DATASETS-TO-COPY-DDN IS ddname

Explanation: This message is informational.

User response: No action is required.

CKZ53507I SYNCDB2-DDN IS ddname

Explanation: This message is informational.

User response: No action is required.

CKZ53508I SQLOUT-DDN IS ddname

Explanation: This message is informational.

User response: No action is required.

CKZ53509I STOP-SRC-DDN IS ddname

Explanation: This message is informational.

User response: No action is required.

CKZ53510I STOP-TRG-DDN IS ddname

Explanation: This message is informational.

User response: No action is required.

CKZ53511I START-SRC-DDN IS ddname

Explanation: This message is informational.

User response: No action is required.

CKZ53512I IDCAMS-DDN IS ddname

Explanation: This message is informational.

User response: No action is required.

CKZ53513I COPY DATA-MOVER PGM IS ppp

Explanation: This message is informational. ppp = program name

User response: No action is required.

CKZ53514I COPY DATA-MOVER FAST-REP IS sss

Explanation: This message is informational. sss = fast replication value

User response: No action is required.

CKZ53515I SIMULATE IS sss

Explanation: This message is informational.

sss = simulation type

Y issue source SQL for LISTDEF objects

A do Y, issue target SQL and call DSS with the NORUN option if Data_Mover is ADRDSSU

N do copies if Data-Mover is ADRDSSU

User response: No action is required.

CKZ53516I OBJECT TRANSLATE, SOURCE=sss, TARGET=ttt

Explanation: This message is informational. sss = source DB2 creator ttt = target DB2 creator

User response: No action is required.

CKZ53517E OBJECT TRANSLATE, ONE OR DUPLICATES

Explanation: This is a user error. One or more objects referred to in an OBJ_XLATE subcommand points to a duplicate. See CKZ53518W messages for duplicate names.

User response: Correct the input and resubmit the job.

CKZ53518W OBJECT TRANSLATE, DUPLICATE OBJECT, ttt, ddd, ooo

Explanation: This is a user error. The object duplicates the object name in a previous OBJ_XLATE command. This will result in duplicates not being used and

therefore objects may not match for the copy. ttt = object type ddd = Source (Target objects may be duplicated) ooo = object name

User response: Correct the input and resubmit the job. This may cause multiple jobs to be submitted with updated LISTDEF cards to eliminate the duplicates.

CKZ53519I XMLSTRING-DDN IS ddname

Explanation: This message is informational.

User response: No action is required.

CKZ53520E XMLSTRING-DDN MUST BE FROM 1 TO 8 CHARACTERS

Explanation: This is a user error.

User response: Correct the input and resubmit the job.

CKZ53521E DSNS-PER-COPY sss, Invalid Length

Explanation: This is a user error. Between 1 and 4 numeric characters must be entered. sss = string entered

User response: Correct the input and resubmit the job.

CKZ53522E DSNS-PER-COPY has an Invalid Value, sss

Explanation: This is a user error. sss = string entered

User response: Correct the input and resubmit the job.

CKZ53523E TARGET LOCATION lll MUST BE FROM 1-128 CHARACTERS

Explanation: This is a user error. Note that the maximum location value in V7 is 16. lll = target DB2 subsystem location

User response: Correct the input and resubmit the job.

CKZ53524E TARGET USERID uid, INVALID LENGTH

Explanation: This is a user error.

User response: Correct the input and resubmit the job.

CKZ53525E TARGET PASSWORD ppp, INVALID LENGTH

Explanation: This is a user error. ppp = password

User response: Correct the input and resubmit the job.

CKZ53526E SERVER IP ipaddr, INVALID LENGTH

Explanation: This is a user error. ipaddr may be IPv4 or IPv6.

User response: Correct the input and resubmit the job.

CKZ53527E Server Port nnn, Invalid Length

Explanation: This is a user error. nnn = decimal number

User response: Correct the input and resubmit the job.

CKZ53528E SERVER-PORT PARM HAS AN INVALID VALUE, sss

Explanation: This is a user error. sss = port string

User response: Correct the input and resubmit the job.

CKZ53529E TARGET VCAT vvv, INVALID LENGTH

Explanation: This is a user error. vvv = VCAT name

User response: Correct the input and resubmit the job.

CKZ53530E DATASETS-TO-COPY-DDN MUST BE FROM 1 TO 8 CHARACTERS

Explanation: This is a user error.

User response: Correct the input and resubmit the job.

CKZ53531E SYNCDB2-DDN MUST BE FROM 1 TO 8 CHARACTERS

Explanation: This is a user error.

User response: Correct the input and resubmit the job.

CKZ53532E STOP-SRC-DDN MUST BE FROM 1 TO 8 CHARACTERS

Explanation: This is a user error.

User response: Correct the input and resubmit the job.

CKZ53533E STOP-TRG-DDN MUST BE FROM 1 TO 8 CHARACTERS

Explanation: This is a user error.

User response: Correct the input and resubmit the job.

CKZ53534E START-SRC-DDN MUST BE FROM 1 TO 8 CHARACTERS

Explanation: This is a user error.

User response: Correct the input and resubmit the job.

CKZ53535E IDCAMS-DDN MUST BE FROM 1 TO 8 CHARACTERS

Explanation: This is a user error.

User response: Correct the input and resubmit the job.

CKZ53536E SIMULATE MUST BE BLANK, Y, N OR A

Explanation: This is a user error.

User response: Correct the input and resubmit the job.

CKZ53537E INVALID SERVER IP ipaddr, REASON rrr

Explanation: This is a user error. ipaddr may be IPv4 or IPv6.

rrr = reason code

- 1 character not decimal digit or a dot
- 2 too many digits with no dot
- 3 no digits for octet
- 4 number of octets not 4
- 11 character not hex digit or colon
- 12 too many digits with no colon
- 13 too many hex pieces
- 14 more than 1 :: or :::
- 15 string > 39 characters
- 16 starts or ends with a single colon
- >20 call IBM Software Support

User response: Correct the input and resubmit the job or if >20, call IBM Software Support.

CKZ53538E SQLOUT-DDN MUST BE FROM 1 TO 8 CHARACTERS

Explanation: This is a user error.

User response: Correct the input and resubmit the job.

CKZ53539E FAST-REP OPERAND vvv IS INVALID

Explanation: This is a user error. vvv = operand string

User response: Correct the input and resubmit the job.

CKZ53540E UNRECOGNIZED KEYWORD IN OBJECT-TRANSLATION COMMAND ccc

Explanation: This is a user error. ccc = command value

User response: Correct the input and resubmit the job.

CKZ53541E ILLEGAL SYNTAX IN
OBJECT-TRANSLATION COMMAND
ccc

Explanation: This is a user error. ccc = command value

User response: Correct the input and resubmit the job.

CKZ53542E INVALID SERVER IP ipaddr, UNABLE
TO DETERMINE IF V4 OR V6
FORMAT

Explanation: This is a user error. ipaddr may be IPv4 or IPv6.

User response: No action is required.

CKZ53543I SERVER IPV6 IS ipaddr

Explanation: This message is informational. ipaddr = IPv6 address

User response: No action is required.

CKZ53544E TARGET VCAT vvv, ILLEGAL VALUE

Explanation: This is a user error.

User response: Correct the input and resubmit the job.

CKZ53545E DSS-COPY-COMMANDS sss, Invalid
Length

Explanation: This is a user error. Between 1 and 4 numeric characters must be entered. sss = string entered

User response: Correct the input and resubmit the job.

CKZ53546E DSS-COPY-COMMANDS has an
Invalid Value, sss

Explanation: This is a user error. sss = string entered

User response: Correct the input and resubmit the job.

CKZ53547I DSNS-PER-COPY is ddd

Explanation: This message is informational. ddd = decimal value entered

User response: No action is required.

CKZ53548I DSS-COPY-COMMANDS is ddd

Explanation: This message is informational. ddd = decimal value entered

User response: No action is required.

CKZ53549W DSNS-PER-COPY is ddd, Out of Range
(1-255), Reverting to Parmlib or Default
Value

Explanation: This message is a warning. ddd = decimal value entered

User response: Correct the input and resubmit the job if MAX_RC=0. If MAX_RC=4, this warning will be ignored.

CKZ53550W DSS-COPY-COMMANDS is ddd, Out of
Range (1-256), Reverting to Parmlib or
Default Value

Explanation: This message is a warning. ddd = decimal value entered

User response: Correct the input and resubmit the job if MAX_RC=0. If MAX_RC=4, this warning will be ignored.

CKZ53551I COPY DATA-MOVER
FCTOPPRCPRIARY is sss

Explanation: This message is informational. sss = ON (specified) or OFF (not specified)

User response: No action is required.

CKZ53554E JOB-TEMPLATE DD ddname IS
INVALID

Explanation: This is a user error. The length of the DD name is invalid.

User response: Correct the input and resubmit the job.

CKZ53555I JOB-TEMPLATE DD PAIR,
INDD=ddname, OUTDD=ddname

Explanation: This message is informational. One message prints for each pair in the JOB-TEMPLATE subcommand of the COPY command.

User response: No action is required.

CKZ53558E COPY PROCESS-DDL-DDN must be
from 1-8 Characters

Explanation: This is an error.

User response: Correct the error and resubmit the job.

CKZ53560E COPY DDL PROCESS-TYPE must be N,
Y, G, X or A

Explanation: This is an error.

User response: Correct the error and resubmit the job.

<hr/> CKZ53561I COPY PROCESS-TYPE is <i>process_type</i> Explanation: This message is informational. ttt=process type User response: No action is required. <hr/>	<hr/> CKZ53568E Invalid Value for GBPCACHE, Must be CHANGED, ALL, SYSTEM, or NONE Explanation: This is an error. User response: Correct the error and resubmit the job. <hr/>
CKZ53562E OBJECT-TRANSLATE VCAT <i>vcat</i> , Invalid Length Explanation: This is an error. User response: Correct the error and resubmit the job. <hr/>	CKZ53569E Invalid Value for LOG, Must be YES or NO Explanation: This is an error. User response: Correct the error and resubmit the job. <hr/>
CKZ53563E DDL-ATTRIBUTE-CHANGE, Duplicate Card, <i>attribute_name</i> , <i>source_string_value</i> , <i>target_string_value</i> , <i>object_change_applies_to</i> , <i>source_mask</i> Explanation: This is an error. User response: Correct the error and resubmit the job. <hr/>	CKZ53570E Invalid Value for TRACKMOD, Must be YES or NO Explanation: This is an error. User response: Correct the error and resubmit the job. <hr/>
CKZ53564E DDL-ATTRIBUTE-CHANGE, One or More Duplicates Explanation: One or more duplicate entries was found for the DDL-ATTRIBUTE-CHANGE parameter. This is an error. User response: Correct the error and resubmit the job. <hr/>	CKZ53571E Invalid Value for CLOSE, Must be YES or NO Explanation: This is an error. User response: Correct the error and resubmit the job. <hr/>
CKZ53565E Unrecognized Keyword in DDL-ATTRIBUTE-CHANGE Command, <i>command_name</i> Explanation: This is an error. The incorrect keyword is listed in the message. User response: Correct the error and resubmit the job. <hr/>	CKZ53572E Invalid Value for DATACAPTURE, Must be NONE or CHANGES Explanation: This is an error. User response: Correct the error and resubmit the job. <hr/>
CKZ53566I DDL-ATTRIBUTE-CHANGE, Attribute= <i>attribute_name</i> , Source Value= <i>source_string_value</i> , Target Value= <i>target_string_value</i> , Applytoobj= <i>applytoobject</i> , Applytomask= <i>source_mask</i> Explanation: This message is informational. User response: No action is required. <hr/>	CKZ53573E Applytotype <i>applytotype</i> is not valid for Statement <i>statement</i> Explanation: This is an error. User response: Correct the error and resubmit the job. <hr/>
CKZ53567E Illegal Syntax in DDL-ATTRIBUTE- CHANGE Command, <i>command_string</i> Explanation: This is an error. User response: Correct the error and resubmit the job. <hr/>	CKZ53574E Invalid Value for <i>attribute</i> , <i>attribute_value</i> Explanation: This is an error. User response: Correct the error and resubmit the job. <hr/>
	CKZ53575E Invalid Length <i>field_value</i> for <i>field_name</i> Explanation: This is an error. User response: Correct the error and resubmit the job. <hr/>
	CKZ53576E DDL-ATTRIBUTE-CHANGE, Source Change Equals Target Change, Attribute= <i>attribute</i> , Src= <i>source_attribute_value</i> , Trg= <i>target_attribute_value</i> Explanation: This is an error. User response: Correct the error and resubmit the job. <hr/>

CKZ53577E **Applytotype** *applytotype_object_type* is not valid for **Attribute** *attribute* using **Value** *source_attribute_value*

Explanation: This is an error.

User response: Correct the error and resubmit the job.

CKZ53578I **COPY LOG-APPLY Command**
command_name is *string_value*

Explanation: The string value of the target input log apply command.

User response: No action is required.

CKZ53579I **COPY LOG-APPLY Command**
command_name is *decimal_value*

Explanation: The decimal value of the target input log apply command.

User response: No action is required.

CKZ53580E *command_name* value must be *decimal_number* **Characters Maximum**

Explanation: The LOG-APPLY parm has an invalid value. The parm must be between 1 and *decimal_number* characters in length and cannot be null.

User response: Correct the error and resubmit the job.

CKZ53581E *command_name* has an Invalid Value, *command_value*

Explanation: The source COPY command has an invalid length.

User response: Correct the error and resubmit the job.

CKZ53582E **COPY_OPTION** *option_name* has an Invalid Value, *value*

Explanation: This message is an error.

User response: Correct the error and resubmit the job.

CKZ53583E Invalid Value for **BUFFERPOOL**, *source_attribute_value*

Explanation: This is an error.

User response: Correct the error and resubmit the job.

CKZ53584E Invalid **BUFFERPOOL** Pairing, **Source** *Pool=source_buffer_pool*, **Target** *Pool=target_buffer_pool*

Explanation: This is an error. The size of the page cannot be changed.

User response: Correct the error and resubmit the job.

CKZ53585I *Source* | *Target* **catalog prefetch databases** *database1, database2, database3, database4, database5.*

Explanation: This message is informational. Up to 5 database names may print with this message. Multiple messages will print if there are more than 5 databases.

User response: No action is required.

CKZ53586E *ddname* **Allocation Error, RC=***return_code*, **RS=***reason_code*, **Error** *Code=dynamic_allocation_return_code*

Explanation: The data set could not be allocated.

User response: Correct the error and resubmit the job.

CKZ53587I **Data Sharing Member,** *ID=data_sharing_ID*, *SSID=source_subsystem_for_this_ID*, *ZPARM=ZPARM_member_for_this_ID*

Explanation: The data sharing member information.

User response: No action is required.

CKZ53588E **PRESMIR operand** *value_entered* is invalid.

Explanation: An incorrect value was enter for the PRESERVE MIRROR subcommand.

User response: Correct the value entered and resubmit the job.

CKZ53591E **END-POINT RBA/LRSN** *value_entered* is invalid, *code* *return_code*

Explanation: An incorrect value was entered for the RBA or LRSN. The *value_entered* displays the invalid value in EBCDIC. The *return_code* listed in the message identified the cause, and can be one of the following:

- 08: Opening "X " was not found.
- 12: Ending single quotation mark was not found.
- 16: The RBA length must be 6 or 10 bytes.
- 20: The RBA length is not 10 bytes when DSNJCNVT has been run.
- 24: The RBA has one or more invalid hexadecimal characters.

User response: Correct the input and resubmit the job.

CKZ53592E **PGM(SRCIMCPY) specified, LOG-APPLY command was not found or is incomplete**

Explanation: This error indicates that the LA-ENABLE command is N or defaulted, or no ZPARM member was entered, or both.

User response: Provide the correct LOG-APPLY

| parameters and resubmit the job.

| **CKZ53593I** The END-POINT for target job log
| apply changed to TO_CURRENT.

| **Explanation:** This message is informational.

| **User response:** No action is required.

| **CKZ53594I** Object mismatch set for *mismatch_type*,
| *return_code*.

| **Explanation:** This message is informational. It displays
| the object mismatch type and return code that was
| specified in the COPY command.

| **User response:** No action is required.

| **CKZ53595E** Illegal syntax in OBJECT-MISMATCH-
| RETURN-CODE command.

| **Explanation:** An invalid parameter was entered in the
| OBJECT-MISMATCH-RETURN code command.

| **User response:** Specify a valid mismatch code and
| return code in the command and resubmit the job.

| **CKZ53596E** OBJECT-MISMATCH-RETURN-CODE
| has one or more duplicates.

| **Explanation:** The same mismatch code was entered
| more than once in the OBJECT-MISMATCH-RETURN-
| CODE command. A mismatch code can be specified
| only once.

| **User response:** Remove the duplicate mismatch code
| from the command and resubmit the job.

CKZ53601E SET ERROR, TARGET-JOB AND
TCP-SERVER-JOB CANNOT BOTH BE
Y

Explanation: This is a user error.

User response: Correct the input and resubmit the job.

CKZ53602I NO XMLSTRING DD FOUND

Explanation: This message is informational.

User response: No action is required.

CKZ53603E XMLSTRING DD MUST BE FROM 1-8
CHARACTERS

Explanation: This is a user error.

User response: Correct the input and resubmit the job.

CKZ53604W ccc SET COMMAND HAS AN
INVALID VALUE, vvv, IGNORED | ccc
SET COMMAND HAS AN INVALID
VALUE, vvv, DEFAULTING TO N

Explanation: This message is a warning. The value
entered for the SET command is not valid. The product
default or the PARMLIB value will be used. ccc =
command name vvv = command value

| This is a user error indicating that a SET command
has an invalid value. The value will be changed to N.

User response: Correct the input and resubmit the job
if MAX_RC=0. If MAX_RC=4, this warning will be
ignored.

CKZ53605I SET COMMAND ccc = vvv

Explanation: This message is informational. ccc =
command name vvv = command value

User response: No action is required.

CKZ53606I NO SQL DD FOUND, NO ALTER SQL
PROCESSING

Explanation: This message is informational.

User response: No action is required.

CKZ53607E SQL-OUTDD MUST BE FROM 1-8
CHARACTERS

Explanation: This is a user error.

User response: Correct the input and resubmit the job.

CKZ53608I SET ttt-STATUS-VALUES USING
DEFAULT STATUS VALUES

Explanation: This message is informational. ttt=
ADVISORY or RESTRICT

User response: No action is required.

CKZ53609I NO DEFAULT SQLID, ALL LISTDEF
OBJECTS MUST BE FULLY
QUALIFIED

Explanation: This message is informational.

User response: No action is required.

CKZ53610E DEFAULT-SQLID MUST BE FROM
1-128 CHARACTERS

Explanation: This is a user error. Note that for DB2 V7
the maximum is 8 characters.

User response: Correct the input and resubmit the job.

CKZ53611E ILLEGAL SYNTAX IN SET
DEFAULT-SQLID COMMAND, vvv

Explanation: This is a user error. vvv = command value

User response: Correct the input and resubmit the job.

CKZ53612I SET COMMAND ccc = vvv

Explanation: This message is informational. ccc = command name vvv = command value

User response: No action is required.

CKZ53613E ILLEGAL VALUE IN SET
ttt-STATUS-VALUES, sss

Explanation: This is a user error. ttt = ADVISORY or RESTRICT sss = bad status value

User response: Correct the input and resubmit the job. Refer to the Command Reference for your version of DB2 for a list of valid values for the ?DISPLAY DATABASE command.

CKZ53614I SET ttt-STATUS-VALUES USING
STATUS VALUE sss

Explanation: This message is informational. One message will print for each status value requested. ttt = ADVISORY or RESTRICT sss = bad status value

User response: No action is required.

CKZ53615I NO ttt STATUS COMMAND ISSUED
PER PARM

Explanation: This message is informational. ttt = ADVISORY or RESTRICT

User response: No action is required.

CKZ53616E sss IS NOT A RECOGNIZED STATUS
VALUE FOR ttt-STATUS-VALUES

Explanation: This is a user error. sss = bad status value ttt = ADVISORY or RESTRICT

User response: Correct the input and resubmit the job. Refer to the Command Reference for your version of DB2 for a list of valid values for the ?DISPLAY DATABASE command.

CKZ53617W ttt-STATUS-VALUES HAS A
DUPLICATE STATUS VALUE FOR sss

Explanation: This is a warning. The duplicate will be ignored if the job runs (MAX-RC=4). ttt = ADVISORY or RESTRICT sss = duplicate status value

User response: Correct the input and resubmit the job.

CKZ53618E ERROR PROCESSING STATUS
VALUES

Explanation: This is a user error. See more detailed messages to determine the exact cause.

User response: Correct the input and resubmit the job.

CKZ53619I DB2_COMMAND_RESPONSE_WAIT
Parm Not in SET Command, Defaulting
to ddd Seconds

Explanation: This message is informational. ddd = decimal value

User response: No action is required.

CKZ53623W ccc SET COMMAND HAS AN
INVALID VALUE, vvv

Explanation: This is a user error indicating a SET command has an invalid value. ccc = the name of the SET command vvv = the invalid value

User response: Correct the SET command value and resubmit the job.

| **CKZ53625E** ddname must be from 1-7 characters

| **Explanation:** The input or output job template member name has an invalid length.

| **User response:** Change the ddname and resubmit the job.

| **CKZ53626I** Server Source IPv4 is ip_address.

| **Explanation:** This message is informational.

| **User response:** No action is required.

| **CKZ53627I** Server Source IPv6 is ip_address.

| **Explanation:** This message is informational.

| **User response:** No action is required.

| **CKZ53628E** SET error, parameter_1 and parameter_2
are incompatible

| **Explanation:** This is a user error. Specify one or the other of these parameters, but not both.

| **User response:** Correct the input and resubmit the job.

| **CKZ53629I** SET command server IPv4 is ipaddr

| **Explanation:** This message is informational.

| **User response:** No action is required.

CKZ53630I SET command server IPv6 is *ip6addr*

Explanation: This message is informational.

User response: No action is required.

CKZ53631E *server_job_1* SET command(s) found in
the *server_job_2* TCPIP job

Explanation: This is a user error. Source server commands cannot be in the target server and vice-versa.

- *server_job_1* = the server job that the SET commands belong to
- *server_job_2* = the server job in which the SET commands were found

User response: Correct the input and resubmit the job.

CKZ53702E SYNCDB2 COMMAND PARSING
ERROR, RC=rrr, RS=sss

Explanation: Target job initialization has failed to read the SYNCDB2 input file using DD CKZIN. rrr = return code sss = reason code

User response: If the user has edited this file, then this probably is a user error. If not, ensure the input file was specified correctly and is the same one the corresponding source job output to the SYNCDB2-DDN.

CKZ53704E MAXIMUM DATASET EXTENSIONS
vzv, INVALID LENGTH

Explanation: The MAXIMUM-EXTENSION value in the SYNCDB2 Command is invalid. vzv = invalid length

User response: If the user has edited this file, then this probably is a user error. If not, ensure the input file was specified correctly and is the same one the corresponding source job output to the SYNCDB2-DDN.

CKZ53705E MAXIMUM DATASET EXTENSIONS
HAS AN INVALID VALUE, vzv

Explanation: The MAXIMUM-EXTENSION value in the SYNCDB2 Command is invalid. vzv = invalid value

User response: If the user has edited this file, then this probably is a user error. If not, ensure the input file was specified correctly and is the same one the corresponding source job output to the SYNCDB2-DDN.

CKZ53709E TARGET-DSN dsname HAS A LENGTH
>44

Explanation: TARGET-DSN in the SYNCDB2 command has a length greater than 44.

User response: If the user has edited this file, then this probably is a user error. If not, ensure the input file was specified correctly and is the same one the corresponding source job output to the SYNCDB2-DDN.

CKZ53710E CISIZE (NNN) MUST BETWEEN 4K
AND 32K

Explanation: The CISIZE is invalid.

User response: Correct the CISIZE of the space and resubmit the source and target jobs.

CKZ53712E PAGE-SIZE (ddd) MUST BE 4, 8, 16 OR
32

Explanation: PAGE-SIZE in the SYNCDB2 command has an invalid length. ddd = invalid value

User response: If the user has edited this file, then this probably is a user error. If not, ensure the input file was specified correctly and is the same one the corresponding source job output to the SYNCDB2-DDN.

CKZ53716E STOP-SPACE (vzv) MUST BE Y OR N

Explanation: STOP-SPACE in the SYNCDB2 command has an invalid value. vzv = invalid value

User response: If the user has edited this file, then this probably is a user error. If not, ensure the input file was specified correctly and is the same one the corresponding source job output to the SYNCDB2-DDN.

CKZ53719E START-SPACE (vzv) MUST BE Y OR N

Explanation: START-SPACE in the SYNCDB2 command has an invalid value. vzv = invalid value

User response: If the user has edited this file, then this probably is a user error. If not, ensure the input file was specified correctly and is the same one the corresponding source job output to the SYNCDB2-DDN.

CKZ53729E DUPLICATE ttt TRANSLATE
COMMANDS

Explanation: Duplicate translate commands found in a SYNCDB2 command. ttt = DBID | TSOB | PSID | ISOB | IXOB

User response: If the user has edited this file, then this probably is a user error. If not, ensure the input file

CKZ53730E • CKZ53803E

was specified correctly and is the same one the corresponding source job output to the SYNCDB2-DDN.

CKZ53730E UNKNOWN IDTYPE ttt

Explanation: Unknown translate commands found in a SYNCDB2 command. ttt = unknown translate string

User response: If the user has edited this file, then this probably is a user error. If not, ensure the input file was specified correctly and is the same one the corresponding source job output to the SYNCDB2-DDN.

CKZ53731E sss OF ddd HAS EXCEEDED MAXIMUM VALUE 65536

Explanation: The parm specified has an invalid value (> 16 bits). Object IDs are limited to 16 bits. sss = SRCPARM | TRGPARM ddd = value of the PARM

User response: If the user has edited this file, then this probably is a user error. If not, ensure the input file was specified correctly and is the same one the corresponding source job output to the SYNCDB2-DDN.

CKZ53733E NON NUMERIC VALUE FOUND IN sss

Explanation: The value specified is not numeric. sss = object ID

User response: If the user has edited this file, then this probably is a user error. If not, ensure the input file was specified correctly and is the same one the corresponding source job output to the SYNCDB2-DDN.

CKZ53734I SYNCDB2 TARGET-SSID IS vvv

Explanation: This message is informational. vvv = the target subsystem

User response: No action is required.

CKZ53735I SYNCDB2 TARGET-DSNAME IS vvv

Explanation: This message is informational. vvv = the VSAM data set name

User response: No action is required.

CKZ53736I SYNCDB2 sss IS = ddd

Explanation: This message is informational. sss = parameter name ddd = decimal parameter value

User response: No action is required.

CKZ53737I SYNCDB2 sss IS = vvv

Explanation: This message is informational. sss = parameter name vvv = parameter value

User response: No action is required.

CKZ53738I SYNCDB2 sss IS = vvv

Explanation: This message is informational. sss = parameter name vvv = parameter value in hex

User response: No action is required.

| CKZ53739I SYNCDB2 SOURCE-DSNAME is | data_set_name

| **Explanation:** This message is informational.

| **User response:** No action is required.

| CKZ53740I SYNCDB2 parameter_name is = | parameter_value_in_decimal

| **Explanation:** This message provides information about parameters.

| **User response:** No action is required.

CKZ53744I SYNCDB2 sss IS = SRC(hhh) TRG(hhh)

Explanation: This message is informational. sss = parameter name hhh = hex parameter value

User response: No action is required.

| CKZ53745E object, parameter, Invalid Length

| **Explanation:** The database or indexspace in the SYNCDB2 member has an invalid length. object = TARGET-TS-DB or TARGET-TS-TS and parameter = value of the parameter.

| **User response:** If the file has been edited, this may be a user error; correct the parameter value. If not, ensure the input file was correctly specified and is the same file that the corresponding source job output to the SYNCDB2-DDN.

CKZ53802E HLQDDDF COMMAND PARSING ERROR, RC=rrr, RS=sss

Explanation: Source job initialization has failed to read the HLQDDDF command. rrr = return code sss = reason code

User response: Correct the input and resubmit the job.

CKZ53803E DIRECTION MUST BE IN OR OUT

Explanation: The HLQDDDF DIR parm must be IN or OUT.

User response: Correct the input and resubmit the job.

CKZ53804E HLQNAME MUST BE 8 CHARACTERS OR LESS

Explanation: The HLQNAME value must be 8 characters or less. This is a data set naming restriction.

User response: Correct the input and resubmit the job.

CKZ53805E HLQDDDF DUPLICATE DEFINITION FOR HLQ=*vvv*

Explanation: The HLQNAME value cannot be specified more than once. *vvv* = duplicate HLQNAME value

User response: Correct the input and resubmit the job.

CKZ53807E REQUIRED KEYWORD, *kkk* MISSING FROM COMMAND

Explanation: The given keyword is required for the HLQDDDF command. *kkk* = missing keyword

User response: Correct the input and resubmit the job.

CKZ53809E BAD FORMAT IN DDNAME *vvv* IN HLQDDDF COMMAND

Explanation: The format of the DDNAME is invalid. *vvv* = DDNAME with the bad format

User response: Correct the input and resubmit the job.

CKZ53810E HLQNAME *nnn* DDNAME *vvv* NOT FOUND

Explanation: The DDNAME given on the HLQDDDF statement is not in the JCL. *nnn* = HLQ name this DD references *vvv* = DDNAME with the bad format

User response: Correct the input and resubmit the job.

CKZ53811E BAD LENGTH IN DDNAME *vvv* IN HLQDDDF COMMAND

Explanation: The length of the DDNAME is invalid. *vvv* = DDNAME with the bad length

User response: Correct the input and resubmit the job.

CKZ53832I HLQDDDF DD DEFINITION DDNAME=*ddname*, DIR=*ddd*

Explanation: This message is informational. *ddd* = IN or OUT

User response: No action is required.

CKZ53901E DATASUBTYPE Command Parsing Error, RC=*return_code*, RS=*reason_code*

Explanation: Source job initialization has failed to read the DATASUBTYPE command.

User response: Correct the input and resubmit the job.

CKZ53902E *value* must be 128 characters or less

Explanation: The DATASUBTYPE field must be less than 128 characters. *value* may be table creator, table name or column name.

User response: Correct the input and resubmit the job.

CKZ53903I DATASUBTYPE *field=value*

Explanation: This gives the parsed value for the given field.

User response: No action is required.

CKZ53904E Illegal Syntax in DATASUBTYPE Command, Field, *field_name*

Explanation: The field has an invalid DB2 value.

User response: Correct the input and resubmit the job.

CKZ53905E SUBTYPE must be S, B or M

Explanation: The FOREIGNKEY value is incorrect.

User response: Correct the input and resubmit the job.

CKZ53906E *field* must be 1 character

Explanation: This DATASUBTYPE field only has one valid length.

User response: Correct the input and resubmit the job.

CKZ54001W CATALOG READ ERROR RETURN, DATA MASKING IS NOW DISABLED

Explanation: This message is a warning.

User response: This is a probable DB2 catalog inconsistency. Correct the error and resubmit the job.

CKZ54002W DATA MASKING REQUESTED FOR AT LEAST ONE TABLE IN DSN=*dsname*. HOWEVER NO OBJECT EXISTS ON THE TARGET, DATA MASKING DISABLED FOR THIS OBJECT

Explanation: This message is a warning. *dsname* = data set name

User response: Either change the MASKCMD or add the table definition on the target subsystem and resubmit the job.

CKZ54003W DATA MASKING REQUESTED FOR AT LEAST ONE TABLE IN DSN=dsname. HOWEVER THE DATASET COULD NOT BE COPIED, DATA MASKING DISABLED FOR THIS OBJECT

Explanation: This message is a warning. dsname=data set name.

User response: Either change the MASKCMD or correct the condition causing the data set not to copy and resubmit the job.

CKZ54004W DATA MASKING REQUESTED FOR TABLE mmm.nnn, COLUMN ccc, HOWEVER, THE COLUMN WAS NOT FOUND IN THE TABLE, ALL DATA MASKING IS NOW DISABLED

Explanation: This message is a warning. Note that data masking for all columns is now disabled. mmm = table creator nnn = table name ccc = column name

User response: Correct the column name in the MASKDEF command and resubmit the job.

CKZ54405I DATA MASKING OKAY FOR TABLE mmm.nnn, COLUMN ccc

Explanation: This message is informational. Data masking information will be passed to the target job to allow the masks to be applied to the data sets involved. mmm = table creator nnn = table name ccc = column name

User response: No action is required.

CKZ54101I DDL Attribute Change, Primary
Obj=primary_object_qualifier_1
primary_object_qualifier_2,
Changed=object_type_changed,
Src=source_attribute_value,
Trg=target_attribute_value

Explanation: The change listed in the message has been made in the generated DDL.

User response: No action is required.

CKZ54102I DDL Attribute Change, Primary
Obj=primary_object_name
first_qualifier.second_qualifier,
Changed=object_type_changed,
Src=source_attribute_value,
Trg=target_attribute_value

Explanation: The change listed in the message has been made in the generated DDL.

User response: No action is required.

CKZ54103I DDL Attribute Change, Primary
Obj=primary_object_name
first_qualifier.second_qualifier.
third_qualifier_in_decimal,
Changed=object_type_changed,
Src=source_name_string,
Trg=target_name_string

Explanation: The change listed in the message has been made in the generated DDL.

User response: No action is required.

CKZ54104I DDL Attribute Change, Primary
Obj=primary_object_name
first_qualifier.second_qualifier.
third_qualifier_in_decimal,
Changed=object_type_changed,
Src=source_attribute_in_decimal,
Trg=target_attribute_in_decimal

Explanation: The change listed in the message has been made in the generated DDL.

User response: No action is required.

CKZ54105I DDL Attribute Change, Primary
Obj=primary_object_name
first_qualifier.second_qualifier.third_qualifier,
Changed=object_type_changed,
Src=source_attribute_in_decimal,
Trg=target_attribute_in_decimal

Explanation: The change listed in the message has been made in the generated DDL.

User response: No action is required.

CKZ54106E DDL Error, No Table spaces being Processed

Explanation: LISTDEF must contain a table space for each set of index spaces for tables in that table space.

User response: Specify all table spaces using a DATABASE or TABLESPACE object definition in LISTDEF and specify ALWAYS-COPY-INDEXSPACES(Y).

CKZ54110I DDL Name Change, Primary
Obj=primary_object_name qualifier,
Fld=field_changed, Src=source_name_value,
Trg=target_name_value

Explanation: The change listed in the message has been made in the generated DDL.

User response: No action is required.

CKZ54111I DDL Name Change, Primary
 Obj=primary_object_name
 first_qualifier.second_qualifier,
 Fld=field_changed, Src=source_name_value,
 Trg=target_name_value

Explanation: The change listed in the message has been made in the generated DDL.

User response: No action is required.

CKZ54112I DDL Name Change, Primary
 Obj=primary_object_name
 first_qualifier.second_qualifier.third_qualifier,
 Fld=field_changed, Src=source_name_string,
 Trg=target_name_string

Explanation: The change listed in the message has been made in the generated DDL.

User response: No action is required.

CKZ54113I DDL Name Change, Primary
 Obj=primary_object_name
 first_qualifier.second_qualifier.third_qualifier,
 Fld=field_changed,
 Src=source_name_in_decimal,
 Trg=target_name_in_decimal

Explanation: The change listed in the message has been made in the generated DDL.

User response: No action is required.

CKZ54120E Table creator.table_name specifies
 DATACAPTURE, but table space
 database.table_space value does not have
 LOG specified

Explanation: This is an error.

User response: Correct the error and resubmit the job.

CKZ54121I DDL Name Change Counts:
 DB=database, TS=table space, TB=table,
 IX=index_name, RL=rel_name, CR=creator

Explanation: This message is informational.

User response: No action is required.

CKZ54122I DDL Attribute Change Counts:
 SG=storage_group, CB=created_by,
 GC=gbp_cache, LG=log,
 PQ=primary_quantity,
 SQ=secondary_quantity, TM=track_mod,
 CL=close, DC=data_capture, BP=bufferpool

Explanation: This message is informational.

User response: No action is required.

CKZ54123E Invalid BUFFERPOOL Pairing for
 Object object_type,
 object_qualifier_1.object_qualifier_2, Source
 Pool=source_bufferpool, Target
 Pool=target_bufferpool

Explanation: This is an error.

User response: Correct the error and resubmit the job.

CKZ54124I Object object_type
 object_qualifier_1.object_qualifier_2, already
 exists on the target, No DDL create will
 be generated

Explanation: This object already exists on the target.

User response: No action is required.

CKZ54125I Source and Target Counts of Existing
 Objects
 DB=source_databases,target_databases
 TS=source_table_spaces,target_table_spaces
 TB=source_tables,target_tables
 IX=source_indexes,target_indexes

Explanation: This message is informational.

User response: No action is required.

CKZ54401E UNABLE TO OPEN CKZLSTDF DD
 FOR LISTDEF PROCESSING

Explanation: This is a user error.

User response: Correct the input and resubmit the job.

CKZ54402E NO RECORDS IN CKZLSTDF DD FOR
 LISTDEF PROCESSING

Explanation: This is a user error.

User response: Correct the input and resubmit the job.

CKZ54403E PARSE ERROR: UNBALANCED
 COMMENTS, SOME DATA MAY BE
 IGNORED

Explanation: This is a user error.

User response: Correct the input and resubmit the job,
 message deleted

CKZ54404E LISTDEF CONVERSION BINARY TO
 DECIMAL ERROR, RC=rrr, RS=sss,
 TOKEN=vvv

Explanation: This is a user error. rrr = return code sss = reason code vvv = the token in error

User response: Correct the input and resubmit the job.

CKZ54404W DATA MASKING REQUESTED FOR TABLE mmm.nnn, COLUMN ccc, HOWEVER, THE COLUMN WAS NOT FOUND IN THE TABLE, ALL DATA MASKING IS NOW DISABLED

Explanation: This message is a warning. Note that data masking for all columns is now disabled. mmm = table creator nnn = table name ccc = column name

User response: Correct the column name in the MASKDEF command and resubmit the job.

CKZ54405E PARTLEVEL MAXIMUM VALUE EXCEEDED, PARTLEVEL = ddd

Explanation: This is a user error. ddd = invalid partition value

User response: Correct the input and resubmit the job.

CKZ54406E NO LISTDEF STATEMENTS TO PROCESS

Explanation: This is a user error.

User response: Correct the LISTDEF input and resubmit the job.

CKZ54407E PARTLEVEL AND RI CAN NOT BOTH BE SPECIFIED FOR OBJECT ccc.nnn IN LIST lll

Explanation: This is a user error. ccc = first object qualifier nnn = second object qualifier lll = list name

User response: Correct the input and resubmit the job.

CKZ54408E ILLEGAL WILDCARD(S) IN INDEXSPACE ccc.nnn IN LIST lll

Explanation: This is a user error. ccc = first object qualifier nnn = second object qualifier lll = list name

User response: Correct the input and resubmit the job.

CKZ54409E ILLEGAL DATABASE NAME ddd IN LIST lll

Explanation: This is a user error. ddd = database name lll = list name

User response: Correct the input and resubmit the job.

CKZ54410E ILLEGAL WILDCARD(S) IN TABLESPACE ddd.nnn IN LIST lll

Explanation: This is a user error. ddd = database name nnn = table space name lll = list name

User response: Correct the input and resubmit the job.

CKZ54411E ILLEGAL WILDCARD IN LIST lll WITH OBJECT TYPE ooo ccc.nnn

Explanation: This is a user error. % and %.% are illegal wildcards. lll = list name ooo = object type ccc = first object qualifier nnn = second object qualifier

User response: Correct the input and resubmit the job.

CKZ54412I TABLESPACE ddd.nnn IN LIST lll INCLUDED

Explanation: This message is informational. ddd = database name nnn = table space name lll = list name

User response: No action is required.

CKZ54413I TABLESPACE ddd.nnn IN LIST lll HAS NOT BEEN INCLUDED, DUPLICATE

Explanation: This message is informational. ddd = database name nnn = table space name lll = list name

User response: No action is required.

CKZ54414I TABLESPACE ddd.nnn IN LIST lll EXCLUDED

Explanation: This message is informational. ddd = database name nnn = table space name lll = list name

User response: No action is required.

CKZ54415I TABLESPACE ddd.nnn IN LIST lll EXCLUDER

Explanation: This message is informational. ddd = database name nnn = table space name lll = list name

User response: No action is required.

CKZ54416I TABLESPACE ddd.nnn IN LIST lll NO EXCLUDES, HAS NO PARTLEVEL, MATCHING OBJECT HAS PARTLEVEL

Explanation: This message is informational. ddd = database name nnn = table space name lll = list name

User response: No action is required.

CKZ54417I TABLESPACE ddd.nnn IN LIST lll NO EXCLUDES PERFORMED

Explanation: This message is informational. ddd = database name nnn = table space name lll = list name

User response: No action is required.

CKZ54418I TABLESPACE ddd.nnn IN LIST lll NO EXCLUDES, HAS PARTLEVEL, MATCHING OBJECT HAS NO PARTLEVEL

Explanation: This message is informational. This

occurs, for example, when a partitioned space is included without the PARTLEVEL specification and a subsequent EXCLUDE uses PARTLEVEL(n). To correct this, specify PARTLEVEL on the first INCLUDE. ddd = database name nnn = table space name lll = list name

User response: No action is required.

CKZ54419I TABLESPACE ddd.nnn IN LIST lll EXCLUDED, ALL PARTS EXCLUDED

Explanation: This message is informational. ddd = database name nnn = table space name lll = list name

User response: No action is required.

CKZ54420I TABLESPACE ddd.nnn IN LIST lll EXCLUDER, ALL PARTS EXCLUDED BY PARTLEVEL NULL

Explanation: This message is informational. See CKZ54409I for the excluded table space. ddd = database name nnn = table space name lll = list name

User response: No action is required.

CKZ54421I TABLESPACE ddd.nnn IN LIST lll HAS EXCLUDED PARTLEVEL ppp

Explanation: This message is informational. ddd = database name nnn = table space name lll = list name ppp = partition number

User response: No action is required.

CKZ54422I TABLESPACE ddd.nnn IN LIST lll NO EXCLUDES, USING PARTLEVEL ppp

Explanation: This message is informational. ddd = database name nnn = table space name lll = list name ppp = partition number

User response: No action is required.

CKZ54423I INDEX ccc.nnn IN LIST lll INCLUDED

Explanation: This message is informational. ccc = index creator nnn = index name lll = list name

User response: No action is required.

CKZ54424I INDEX ccc.nnn IN LIST lll HAS NOT BEEN INCLUDED, DUPLICATE

Explanation: This message is informational. ccc = index creator nnn = index name lll = list name

User response: No action is required.

CKZ54425I INDEX ccc.nnn IN LIST lll EXCLUDED

Explanation: This message is informational. ccc = index creator nnn = index name lll = list name

User response: No action is required.

CKZ54426I INDEX ccc.nnn IN LIST lll EXCLUDED

Explanation: This message is informational. ccc = index creator nnn = index name lll = list name

User response: No action is required.

CKZ54427I INDEX ccc.nnn IN LIST lll NO EXCLUDES, HAS NO PARTLEVEL, MATCHING OBJECT HAS PARTLEVEL

Explanation: This message is informational. ccc = index creator nnn = index name lll = list name

User response: No action is required.

CKZ54428I INDEX ccc.nnn IN LIST lll NO EXCLUDES PERFORMED

Explanation: This message is informational. ccc = index creator nnn = index name lll = list name

User response: No action is required.

CKZ54429E LISTDEF TOKEN ttt NOT COMPLETED

Explanation: This message is informational. ttt = name of the Listdef token

User response: Correct the input and resubmit the job.

CKZ54430I INDEX ccc.nnn IN LIST lll NO EXCLUDES, HAS PARTLEVEL, MATCHING OBJECT HAS NO PARTLEVEL

Explanation: This message is informational. ccc = index creator nnn = index name lll = list name

User response: No action is required.

CKZ54431I INDEX ccc.nnn IN LIST lll EXCLUDED, ALL PARTS EXCLUDED

Explanation: This message is informational. ccc = index creator nnn = index name lll = list name

User response: No action is required.

CKZ54432I INDEX ccc.nnn IN LIST lll EXCLUDER, ALL PARTS EXCLUDED

Explanation: This message is informational. See CKZ54421I for the excluded table space. ccc = index creator nnn = index name lll = list name

User response: No action is required.

CKZ54433I INDEX ccc.nnn IN LIST lll HAS EXCLUDED PARTLEVEL ppp

Explanation: This message is informational. ccc = index creator nnn = index name lll = list name ppp = partition number

User response: No action is required.

CKZ54434I INDEX ccc.nnn IN LIST lll NO EXCLUDES, USING PARTLEVEL ppp

Explanation: This message is informational. ccc = index creator nnn = index name lll = list name ppp = partition number

User response: No action is required.

CKZ54435I tt OBJECT ooo WAS NOT TRANSLATED

Explanation: This message is informational. It appears when WARN-IF-OBJECT-NOT-TRANSLATED(N). Indexspace (IS) may be specified, however, no message will appear if not found. The IX object type message will be printed in its place. tt = object type [DB, TS, TB, CR, IX] ooo = object name

User response: No action is required.

CKZ54436W TABLESPACE ddd.nnn IN LIST lll HAS SPECIFIED A PARTLEVEL ppp THAT EXCEEDS THE NUMBER OF PARTITIONS

Explanation: This is a user error. The number of partitions in the table space is less than the specified PARTLEVEL. ddd = database name nnn = table space name lll = list name ppp = partition number

User response: Correct the input and resubmit the job.

CKZ54437W INDEX ccc.nnn IN LIST lll HAS SPECIFIED A PARTLEVEL ppp THAT EXCEEDS THE NUMBER OF PARTITIONS

Explanation: This is a user error. The number of partitions in the index is less than the specified PARTLEVEL. ccc = index creator nnn = index name lll = list name ppp = partition number

User response: Correct the input and resubmit the job.

CKZ54438W tt OBJECT ooo WAS NOT TRANSLATED

Explanation: This message is a warning. It appears when WARN-IF-OBJECT-NOT-TRANSLATED(Y). Indexspace (IS) may be specified, however, no message will appear if not found. The IX object type message

will be printed in its place. tt = object type [DB, TS, TB, CR, IX] ooo = object name

User response: Verify that all object translations were done correctly. See CKZ56904I in CKZLOG to determine which masks were used for each translation.

CKZ54439I TABLESPACE ddd.nnn IN LIST lll HAS OVERLAYED PARTLEVEL (?) WITH PARTLEVEL (ALL)

Explanation: This message is informational. ddd = database name nnn = table space name lll = list name

User response: No action is required.

CKZ54440I TABLESPACE ddd.nnn IN LIST lll HAS INCLUDED PARTLEVEL ppp

Explanation: This message is informational. ddd = database name nnn = table space name lll = list name ppp = partition number

User response: No action is required.

CKZ54441I INDEX ccc.nnn IN LIST lll HAS OVERLAYED PARTLEVEL (?) WITH PARTLEVEL (ALL)

Explanation: This message is informational. ccc = index creator nnn = index name lll = list name

User response: No action is required.

CKZ54442I INDEX ccc.nnn IN LIST lll HAS INCLUDED PARTLEVEL ppp

Explanation: This message is informational. ccc = index creator nnn = index name lll = list name ppp = partition number

User response: No action is required.

CKZ54443I TABLE ccc.nnn IN LIST lll HAS NOT BEEN INCLUDED, DUPLICATE

Explanation: This message is informational. ccc = table creator nnn = table name lll = list name

User response: No action is required.

CKZ54444I TABLE ccc.nnn IN LIST lll HAS BEEN INCLUDED

Explanation: This message is informational. ccc = table creator nnn = table name lll = list name

User response: No action is required.

CKZ54445I CLONED SPECIFIED WITH TABLE SPECIFICATION ccc.nnn IN LIST III, IGNORED (V9 AND HIGHER).

Explanation: This message is informational. ccc = table creator nnn = table name III = list name

User response: No action is required.

CKZ54446I CLONED SPECIFIED WITH V8 OR LOWER, IGNORED FOR LIST III

Explanation: This message is informational. III = list name

User response: No action is required.

CKZ54447E XML SPECIFIED IN LIST III

Explanation: This is an error. XML spaces are tolerated but not yet supported. III = LIST name

User response: Correct the LISTDEF statement and resubmit the source job.

CKZ54448W XML COLUMN ccc IN TABLE mmm.nnn IN LIST III, XML SPACES NOT YET PROCESSED

Explanation: This is an error. XML spaces are tolerated but not yet supported. ccc = column name mmm = creator nnn = name III = LIST name

User response: Correct the LISTDEF statement and resubmit the source job.

CKZ54450I PROCESSING LISTDEF aaa bbb ccc ddd eee fff

Explanation: This message is informational. aaa = LISTDEF list name bbb = object specification (table space or indexspace) ccc = object type ddd = other specifications such as LOB or RI eee = first object qualifier fff = second object qualifier

User response: No action is required.

CKZ54451I CATALOG READ COUNTS BY OBJECT: DB=aaa TS=bbb TP=ccc TB=ddd IX=eee IP=fff RL=ggg CL=hhh XR=iii

Explanation: This message is informational. aaa = database bbb = table space ccc = table space part ddd = table eee = index fff = index part ggg = rels hhh = columns iii = auxiliary tables

User response: No action is required.

CKZ54452E UNABLE TO SORT LDF ENTRIES, RS=sss

Explanation: This is an internal error. sss = reason code

User response: Call IBM Software Support.

CKZ54453E NO OBJECTS SELECTED

Explanation: This is a user error. LISTDEF processing has not resulted in any objects being selected.

User response: Correct the input and resubmit the job.

CKZ54454I SIMULATE WITHOUT ALLOCATIONS COMPLETED

Explanation: This message is informational.

User response: No action is required.

CKZ54455I TABLESPACE ddd.nnn IN LIST III HAS NOT BEEN INCLUDED, CLONE yyy

Explanation: This message is informational. ddd = database name nnn = table space name III = list name yyy = YES or NO

User response: No action is required.

CKZ54456I INDEXSPACE ddd.nnn IN LIST III HAS NOT BEEN INCLUDED, CLONE yyy

Explanation: This message is informational. ddd = database name nnn = indexspace name III = list name yyy = YES or NO

User response: No action is required.

CKZ54457I COPY REQUESTS TO BE COMPLETED BY THE USER, TABLESPACES ttt, INDEXSPACES iii

Explanation: This message is informational. ttt = count of table spaces iii = count of indexspaces

User response: No action is required.

CKZ54459W DATA-MASKING(Y) BUT LISTDEF III DOES NOT HAVE THE RI ATTRIBUTE

Explanation: This is a user error. III = list name

User response: Set INCLUDE-ALL-RI(Y) or set the RI indicator in all LISTDEF statements. If this warning isn't needed, set WARN-ON-INCOMPLETE-RI(N). Correct the input and resubmit the job.

CKZ54460I DDL-ATTRIBUTE-CHANGE command generated from OBJECT-TRANSLATE command for object=*object_type_changed*, source=*value_of_source_object*, target=*value_of_target_object*

Explanation: This message is informational.

User response: No action is required.

CKZ54461E One or more LISTDEF INCLUDE TABLESPACE statements use an object spec other than DATABASE or TABLESPACE with source catalog prefetch enabled

Explanation: This is a user error.

User response: Either turn off catalog prefetch on the source catalog, or specify all table space LISTDEF statements with an object spec of DATABASE or TABLESPACE, and resubmit the source job.

CKZ54463W DDL output, PGM parm is not NONE, no copies to be performed.

Explanation: The DDL PROCESS-TYPE parameter is set to X. Copies will not be performed.

User response: Rerun the source job using PROCESS-TYPE=X to execute the DDL. Then rerun the source job a second time to perform the copies.

CKZ54501E LENGTH EXCEEDED, STRING sss

Explanation: This is a user error. sss = string with bad length

User response: Correct the input and resubmit the job.

CKZ54502E FIELD DELIMITER ERROR, STRING sss

Explanation: This is a user error. sss = string with the error

User response: Correct the input and resubmit the job.

CKZ54503E ILLEGAL CHARACTER IN STRING sss

Explanation: This is a user error. sss = string with the error

User response: Correct the input and resubmit the job.

CKZ54504W ONE OR MORE DATASETS MAY NEED TO BE SYNCED TO THE TARGET DB2, NO SYNCDB2 DD IS AVAILABLE

Explanation: This is a warning.

User response: Add the SYNCDB2 DD to the source job and resubmit the job.

CKZ54505I XMLSTRINGS OUTPUT OPEN OK FOR DDNAME ddname

Explanation: This message is informational.

User response: No action is required.

CKZ54506I ddd XMLSTRING STATEMENT(S) HAVE BEEN WRITTEN FOR TARGET SUBSYSTEM ssss

Explanation: This message is informational. ddd = decimal number of strings added to the catalog ssss = DB2 subsystem

User response: No action is required.

CKZ54507E XMLSTRING OPEN FAILED FOR DDNAME ddname, RC=rrr, RS=sss

Explanation: This is a user error. rrr = return code sss = reason code

User response: Correct the data set and resubmit the job.

CKZ54508W TARGET SUBSYSTEM ssss DOES NOT SPECIFY AN XMLSTRINGS DD. XML TABLESPACES MAY NOT BE ACCESSIBLE ON THE TARGET

Explanation: This is a user error. ssss = target subsystem

User response: Correct the data set and resubmit the job.

CKZ54508W TARGET SUBSYSTEM ssss DOES NOT SPECIFY AN XMLSTRINGS DD. XML TABLESPACES MAY NOT BE ACCESSIBLE ON THE TARGET

Explanation: This is a user error. ssss = target subsystem

User response: Correct the data set and resubmit the job.

CKZ54510E statement

Explanation: This is a user error. statement = the Listdef statement in error

User response: Correct the input and resubmit the job.

CKZ54511E THE FOLLOWING LISTDEF STATEMENT HAS BEEN TERMINATED BY AN ERROR

Explanation: This is a user error.

User response: Correct the input and resubmit the job.

CKZ54512E THE FOLLOWING LISTDEF STATEMENT PRECEDED THE LISTDEF STATEMENT IN ERROR

Explanation: This is a part of a multiple messages describing the Listdef error.

User response: Correct the input and resubmit the job.

CKZ54513I statement

Explanation: This message is informational. statement = the Listdef statement

User response: No action is required.

CKZ54514I TOKEN(S) EXPECTED INSTEAD OF sss, ttt

Explanation: This message is informational. sss = the bad token ttt = a list of one or more tokens that were expected

User response: No action is required.

CKZ54515I TARGET SUBSYSTEM ttt IS THE SAME AS THE SOURCE SUBSYSTEM, NO XMLSTRING PROCESSING REQUIRED

Explanation: This message is informational. ttt = target subsystem

User response: No action is required.

CKZ54521I BEGIN SOURCE TABLESPACE DATASET REPORT

Explanation: This message is informational.

User response: No action is required.

CKZ54522I TABLESPACE OBJECT COUNT: ddd, DATASET COUNT: ddd

Explanation: This message is informational. ddd = decimal count

User response: No action is required.

CKZ54523I BEGIN SOURCE INDEXSPACE DATASET REPORT

Explanation: This message is informational.

User response: No action is required.

CKZ54524I INDEXSPACE OBJECT COUNT: ddd, DATASET COUNT: ddd

Explanation: This message is informational. ddd = decimal count

User response: No action is required.

CKZ54525I END SOURCE TABLESPACE/ INDEXSPACE DATASET REPORT

Explanation: This message is informational.

User response: No action is required.

CKZ54527I DSN=dsname LIST=bbb TYPE=ccc OBJ=ddd eee.fff

Explanation: This message is informational. bbb = LIST name ccc = object specification (table space or indexspace) ddd = object type eee = first object qualifier fff = second object qualifier

User response: No action is required.

CKZ54528I DSN=dsname

Explanation: This message is informational.

User response: No action is required.

CKZ54529I ttt tt ccc.nnn

Explanation: This message is informational. ttt = TABLE or AUXTB tt = table type: blank if base with no clone; B - base with a clone; C - clone, M - materialized query table ccc = table creator nnn = table name

User response: No action is required.

CKZ54531I CKZC OUTPUT OPEN OK FOR DDNAME ddname

Explanation: This message is informational. CKZC is the COPY parameter DATASETS-TO-COPY-DDN data set.

User response: No action is required.

CKZ54532W CKZC OUTPUT OPEN FAILED FOR DDNAME ddname, RC=rrr, RS=sss, USE CKZPRINT TO GET DATASET NAMES

Explanation: This message is a warning that the DDName pointed to by the COPY parameter DATASETS-TO-COPY-DDN could not be opened. rrr = return code sss = reason code

User response: Correct the CKZC data set and resubmit the job or get the data sets to copy from CKZPRINT.

CKZ54541I CKZS OUTPUT OPEN OK FOR DDNAME ddname

Explanation: This message is informational. CKZS is the COPY parameter SLOUT-DDN data set.

User response: No action is required.

CKZ54542E CKZS OUTPUT OPEN FAILED FOR DDNAME ddname, RC=rrr, RS=sss

Explanation: This message is a warning that the DDName pointed to by the COPY parameter SQLOUT-DDN could not be opened. rrr = return code sss = reason code

User response: Correct the CKZC data set and resubmit the job.

CKZ54544I OBJID ooo=hhh FOR ttt nnn.sss PART(ppp) IS NOT CHANGING, NO XLATE COMMAND REQUIRED

Explanation: This message is informational. ooo = object ID field name hhh = hexadecimal object ID value ttt = data set type, TS or IS nnn = database name sss = space name ppp = partition number

User response: No action is required.

CKZ54551I SQL OUTPUT OPEN OK FOR DDNAME ddname

Explanation: This message is informational.

User response: No action is required.

CKZ54552I ddd SQL STATEMENT(S) HAVE BEEN WRITTEN FOR TARGET SUBSYSTEM ssss

Explanation: This message is informational. ddd = decimal number ssss = DB2 subsystem

User response: No action is required.

CKZ54553E SQL OUTPUT OPEN FAILED FOR DDNAME DDNAME, RC=rrr, RS=sss

Explanation: This message is a warning that the DDName pointed to by the COPY parameter SQLOUT-DDN could not be opened. rrr = return code sss = reason code

User response: Correct the data set and resubmit the job.

CKZ54554W TARGET SUBSYSTEM ssss DOES NOT SPECIFY A CKZQ DD FOR SQL OUTPUT. ONE OR MORE SQL STATEMENTS CAN NOT BE PROCESSED ON THE TARGET

Explanation: This message is a warning that the SQLOUT-DDN parameter on the COPY command could not be found. sss = DB2 subsystem

User response: Correct the input and resubmit the job.

CKZ54556W ALTER TABLE ALTER IDENTITY COLUMN FOR DB2 V7 IS NOT AVAILABLE, TARGET TABLE ssss.ccc.nnn CANNOT HAVE IDENTITY COLUMN III SYNCHRONIZED WITH THE COPIED SOURCE TABLE

Explanation: This message is a warning. ssss = target DB2 subsystem ccc = table creator nnn = table name III = identity column name

User response: If synchronization is required, synchronize source and target identity columns values before attempting to copy.

CKZ54557W TARGET TABLE ssss.ccc.nnn HAS AN IDENTITY COLUMN III WITH A NON ZERO CACHE VALUE. THIS MAY RESULT IN INSERT OR UPDATE FAILURES DUE TO DUPLICATE IDENTITY COLUMN VALUES, AS THE MAXASSIGNEDVALS ARE SAME ON THE SOURCE AND THE TARGET TABLES

Explanation: This message is a warning. ssss = target DB2 subsystem ccc = table creator nnn = table name III = identity column name

User response: If synchronization is required, synchronize source and target identity columns values before attempting to copy.

| **CKZ54558I** *parameter_value*

| **Explanation:** This message contains hard-coded log apply parms.

| **User response:** No action is required.

CKZ54589W ONE OR MORE OBJECT DATASETS COPIED WITHOUT A TARGET OBJECT, NO IDCAMS_DDN WAS AVAILABLE

Explanation: This is a user error.

User response: Correct the input and resubmit the job.

CKZ54595I OUTPUT OPEN OK FOR DDNAME ddname

Explanation: This message is informational.

User response: No action is required.

CKZ54596W OUTPUT OPEN FAILED FOR DDNAME ddname, RC=rrr, RS=sss

Explanation: This is a user error. rrr = return code sss = reason code

User response: Correct the input and resubmit the job.

CKZ54597W PGM IS NONE HOWEVER, NO STOP-SOURCE DD IS AVAILABLE FOR THE DB2 STOP COMMANDS ON THE SOURCE SUBSYSTEM

Explanation: This is a user error.

User response: Correct the input and resubmit the job.

CKZ54598W PGM IS NONE HOWEVER, NO STOP-TARGET DD IS AVAILABLE FOR THE DB2 STOP COMMANDS ON THE TARGET SUBSYSTEM

Explanation: This is a user error.

User response: Correct the input and resubmit the job.

CKZ54599W PGM IS NONE HOWEVER, NO START-SOURCE DD IS AVAILABLE FOR THE DB2 START COMMANDS ON THE SOURCE SUBSYSTEM

Explanation: This is a user error.

User response: Correct the input and resubmit the job.

CKZ54601I NO DB2 VSAM DATASET WRITES, SET SCAN-ONLY (Y) IS IN EFFECT

Explanation: This message is informational.

User response: No action is required.

CKZ54602I Total page set IO: reads=*reads_in_hex*, writes=*writes_in_hex*

Explanation: This message is informational.

User response: No action is required.

CKZ54603I TOTAL PAGE SET IO: PGRK=*aaa*, PGWK=*bbb*, PTT=*ccc*, PGRW=*ddd*, PTBK=*eee*

Explanation: This message is informational. *aaa* = page reads, *ok bbb* = page writes, *ok ccc* = points *ddd* = page reads, total *eee* = page writes, total

User response: No action is required.

CKZ54604I NO XML UPDATES PERFORMED, SET SCAN-ONLY(Y) IS IN EFFECT

Explanation: This message is informational.

User response: No action is required.

CKZ54605E XMLSTRING CONVERSION BINARY TO DECIMAL ERROR, RC=*rrr*, RS=*sss*, TOKEN=*vvv*

Explanation: This is a user error. *rrr* = return code *sss* = reason code *vvv* = the token in error

User response: Correct the input and resubmit the job.

CKZ54606E XMLSTRING FORMAT ERROR, STRINGLN NOT FOUND, TOKEN=*vvv*

Explanation: This is a user error. *vvv* = the token in error

User response: Correct the input and resubmit the job.

CKZ54607E XMLSTRING FORMAT ERROR, STRING LENGTH > 1000, TOKEN = *vvv*

Explanation: This is a user error. The length is constrained by the length of the XMLSTRINGS catalog column STRING. *vvv* = the token in error

User response: Correct the input and resubmit the job.

CKZ54608E XMLSTRING FORMAT ERROR, UNEXPECTED LINE, TOKEN = *vvv*

Explanation: This is a user error. *vvv* = the token in error

User response: Correct the input and resubmit the job.

CKZ54609E UNABLE TO OPEN *ddname* DD FOR XMLSTRING SOURCE PROCESSING, RC=*rrr*, RS=*sss*

Explanation: This is a user error. *rrr* = return code *sss* = reason code

User response: Correct the data set and resubmit the job.

CKZ54610I *ddd* *iii* XMLSTRINGS PROCESSED

Explanation: This message is informational. It is the count of the STRING IDs read. *ddd* = number of strings processed *iii* = source or target

User response: No action is required.

CKZ54611W ALL TARGET DATASET CHANGES COMPLETE, WITH ERRORS, TSOK *aaa*, ISOK *bbb*, TSERR *ccc*, ISERR *ddd*

Explanation: This message is a warning. *aaa* = count of table spaces *ok bbb* = count of indexspaces *ok ccc* = count of table spaces with errors *ddd* = count of indexspaces with errors

User response: Determine the cause of the error(s) and resubmit the job.

CKZ54612I ALL TARGET DATASET sss
COMPLETED WITHOUT ERRORS,
TABLESPACES aaa, INDEXSPACES bbb

Explanation: This message is informational. sss = Changes or Scans aaa = count of table spaces ok bbb = count of indexspaces ok

User response: No action is required.

CKZ54613E ERROR READING iii
XMLSTRINGS,RC=rrr, RS=sss

Explanation: This is a probable DB2 error. iii = source or target rrr = return code sss = reason code

User response: Correct the error and resubmit the job.

CKZ54614I ddd iii XMLSTRINGS LINES READ

Explanation: This message is informational. ddd = decimal number of lines read from XMLSTRING DD iii = source

User response: No action is required.

CKZ54615I ddd iii CATALOG XMLSTRINGS
READ

Explanation: This message is informational. ddd = decimal number of strings read from XMLSTRINGS catalog table iii = target

User response: No action is required.

CKZ54616I iii CATALOG XMLSTRING ID sss IS A
DUPLICATE, DISCARDED

Explanation: This message is informational. iii = target sss = hex string ID

User response: No action is required.

CKZ54617I TARGET CREATE REQUIRED FOR
SOURCE XMLSTRING ID sss, STRING
ssss

Explanation: This message is informational. The CKZCRXML DD must be available when this message is output. sss = string ID ssss = string

User response: No action is required.

CKZ54618I PREEXISTING XMLSTRING IDS
MAPPED: SRC=sss, TRG=ttt, ssss

Explanation: This message is informational. Source string IDs are passed from the source job in a data set. Target string IDs are read from the target catalog by the target job. sss = hex source string ID ttt = hex target string ID ssss = first 80 bytes of the string value

User response: No action is required.

CKZ54619I ddd TARGET XMLSTRING IDS
CREATED

Explanation: This message is informational. ddd = decimal number

User response: No action is required.

CKZ54621I NO SQL WILL BE EXECUTED, SET
SCAN-ONLY(Y) IS IN EFFECT

Explanation: This message is informational.

User response: No action is required.

CKZ54622I ddd SQL ALTER STATEMENT(S)
EXECUTED SUCCESSFULLY

Explanation: This message is informational. ddd = decimal number of statements

User response: No action is required.

CKZ54623I CURRENTLY EXISTING XMLSTRING
IDS MAPPED: SRC=sss, TRG=ttt, ssss

Explanation: This message is informational. Source string IDs are passed from the source job in a data set. Target string IDs include those read from the target catalog and newly added string values that do not exist in the target catalog but do exist in the source catalog. sss = hex source string ID ttt = hex target string ID ssss = first 80 bytes of the string value

User response: No action is required.

CKZ54624E ERROR CREATING XMLSTRINGS,
RC=rrr, RS=sss

Explanation: This is a probable DB2 error. rrr = return code sss = reason code

User response: Correct the error and resubmit the job.

CKZ54627I Nodes found=*number_of_XML*
nodes_found_in_hex, strings
changed=*number_of_XML string*
IDs_changed_in_hex

Explanation: This message provides information about XML processing.

User response: No action is required.

CKZ54628I XML pages printed=*number_printed*

Explanation: This message provides information about the number of XML pages that were printed.

User response: No action is required.

CKZ54641I Begin completion status report.

Explanation: This message precedes a report that provides information about the data sets processed in the target job. The following columns are provided in the report:

TARGET DATASET

The name of the target data set.

RC The 2-digit return code from data set processing.

SPACE TYPE

The space type: IS (index space) or TS (table space).

PAGES The total number of pages in the data set.

LOG PAGES CHANGED

The number of pages that had one or more log records applied.

DATA MASKING PAGES CHANGED

The number of pages that had one or more data masking changes applied.

VSAM READS

The total number of VSAM reads, including zero pages.

VSAM WRITES

The total number of changed pages.

IO ERR

This column contains Y if an I/O error occurred.

VSAM ERR

This column contains Y if a VSAM error occurred.

EXT FMT

This column contains Y if the page set has a 10-byte RBA/LRSN, or a blank if the page set has a 6-byte RBA/LRSN.

SECS The elapsed time to process the data set in seconds.

User response: No action is required.

CKZ54701W TARGET SS ssss, OBJECT NOT IN DB2 CATALOG, tt ddd.sss, CAN NOT BE COPIED PER PARMS

Explanation: This message is a warning. ssss = DB2 subsystem tt = object type (TS or IS) ddd = database name sss = space name

User response: Change COPY-IF-NO-DB2-TARGET-OBJECTS to Y or add the object to the target subsystem and resubmit the job.

CKZ54702W TARGET SS ssss, OBJECT IN DB2 CATALOG, tt ddd.sss, WITH DEFINE NO, CANNOT BE COPIED

Explanation: This message is a warning. ssss = DB2 subsystem tt = object type (TS or IS) ddd = database name sss = space name

User response: Correct the error and resubmit the job.

CKZ54703W TARGET SS ssss, OBJECT IN DB2 CATALOG, tt ddd.sss, BUT NO PARTS FOUND, DB2 CATALOG PROBLEM, CANNOT BE COPIED

Explanation: This message is a warning. ssss = DB2 subsystem tt = object type (TS or IS) ddd = database name sss = space name

User response: Correct the catalog and resubmit.

CKZ54704I TARGET SS ssss, OBJECT NOT IN DB2 CATALOG, tt ddd.sss, PARMS ALLOW COPY

Explanation: This message is informational. ssss = DB2 subsystem tt = object type (TS or IS) ddd = database name sss = space name

User response: No action is required.

CKZ54705E SOURCE TABLESPACE ddd.sss AND TARGET TABLESPACE ddd.sss HAVE AN UNEQUAL NUMBER OF TABLES, OBIDS CANNOT BE MATCHED

Explanation: This message indicates that the table OBIDs in the source and target table spaces cannot be mapped and thus cannot be changed. ddd = database name sss = space name

User response: Correct the table space definitions to match and resubmit the job.

CKZ54706E tt TABLE ccc.nnn CANNOT NOT BE MATCHED, OBIDS CANNOT BE TRANSLATED

Explanation: This message indicates that the table OBIDs in the source and target table spaces cannot be mapped and thus cannot be changed. The table names must be the same on source and target or must be mapped with OBJ_XLATE. tt = source or target ccc = creator nnn = table name

User response: Change the table names to match or use OBJ_XLATE to map the names and resubmit the job.

CKZ54707I SOURCE TABLE ccc.nnn IN
TABLESPACE ddd.sss AND TARGET
TABLE ccc.nnn IN TABLESPACE
ddd.sss ARE PAIRED FOR OBID
TRANSLATION

Explanation: This message is informational. ccc = creator nnn = table name ddd = database name sss = space name

User response: No action is required.

CKZ54708W TARGET INDEX REBUILD REQUIRED
FOR INDEX ccc.nnn

Explanation: This message is a warning. The LIMITKEYS do not match between the source and target indexes. If MAX_RC=4, the index will be copied. ccc = index creator nnn = index name

User response: The user must rebuild the index on the target.

CKZ54709I TARGET SS ssss, tt OBJECT WITH
SOURCE DSN=dsname AND TARGET
DSN=dsname, PARMS SPECIFIED
PGM NONE, USER IS RESPONSIBLE
FOR THE COPY TO THE TARGET

Explanation: This message is informational. ssss = DB2 subsystem tt = object type (TS or IS)

User response: No action is required.

CKZ54710W TARGET SS tttt, tt SOURCE OBJECT
ddd.ttt, PARTITION ppp SOURCE
OBJECT PARTED, TARGET NOT
PARTED, WILL NOT BE COPIED

Explanation: This message is a warning that no copy will be attempted. tttt = target DB2 subsystem tt = object type (TS or IS) ddd = source database name ttt = source space name ppp = source partition number

User response: Correct the object(s) and resubmit the job.

CKZ54711W ONE OR MORE DATASETS HAVE
BEEN EXCLUDED FROM COPY DUE
TO OBJECT ATTRIBUTE NON
COMPARE OR STATUS FAILURE

Explanation: This message is a warning.

User response: Correct the error and resubmit the job.

CKZ54712W DATASET ddd HAS FAILED OBJECT
MATCH CHECKING, WILL BE
COPIED PER ALLOW-COPY-ON-
MISMATCH(Y) TO SUBSYSTEM tttt

Explanation: This message is a warning. The copy of this data set will be attempted in spite of the object

mismatch. For the copy to proceed, MAX_RC must be set to 4. If not, the job will end on the warning. ddd = source data set name tttt = target subsystem

User response: Closely check the viability of the target space with the mismatch after the copy completes.

CKZ54713W tt DATASET ddd IS NOT CATALOGED
ON THE SOURCE SYSTEM, CANNOT
BE COPIED

Explanation: This message is a warning that this data set is not cataloged on the source z/OS system. tt = data set type (TS or IS) ddd = source data set name

User response: The object exists in the catalog but the data set may be migrated or have been deleted. If migrated, recall the data set and rerun the job. If deleted, use the Exclude keyword to prevent the object from being considered for copy.

CKZ54714W tt DATA SET ddd IS NOT
CATALOGED ON THE TARGET
SYSTEM, AND THE OBJECT EXISTS,
WILL BE COPIED TO THE TARGET
SUBSYSTEM USING CKZ 5TH LEVEL
QUALIFIER, F0001

Explanation: This message is a warning that this data set is not cataloged on the target z/OS system. tt = data set type (TS or IS) ddd = source data set name

User response: Run the target job to change OBIDs and reset log RBAs. Recall migrated data set(s) or use IDCAMS to allocate missing data set(s), then run IDCAMS using the IDCAMS input statements to delete the dummy data set(s) and rename the copied data set(s).

CKZ54715W TARGET SS ssss, tt OBJECT WITH
SOURCE DSN=dsname, NO TARGET
OBJECTS FOUND AND NO DEFAULT
VCAT SPECIFIED, COPY CANNOT BE
PERFORMED

Explanation: This message is a warning. ssss = DB2 subsystem tt = object type (TS or IS)

User response: Correct the error and resubmit the job.

CKZ54716W TARGET SS ssss, TT OBJECT ddd.nnn,
TARGET VCAT IS RESERVED FOR
DB2 SYSTEM DATASETS, CANNOT
COPY.

Explanation: This message is a warning. ssss = DB2 subsystem tt = object type (TS or IS) ddd = database name sss = space name

User response: Correct and resubmit the job.

CKZ54718W tt DATASET dsname PAGE SIZE IS ddd
AND CISIZE IS eee

Explanation: This message is a warning and is output only when FUZZY-COPY is Yes. tt = object type (table space or index space) dsname = data set name ddd = page size in decimal bytes eee = VSAM CISIZE size in decimal bytes

User response: Submit the source job with FUZZY-COPY(N) or change the CISIZE of the data set to equal the page size.

CKZ54719W tt dd.ss STATUS IS sss, SUBSYSTEM
ssss

Explanation: This message is informational. tt = space type (table space or index space) dd = database ss = space name sss = status value ssss = source or target subsystem name

User response: Determine the cause of the status condition and correct if necessary.

CKZ54720W tt dd.ss.pp STATUS IS sss, SUBSYSTEM
ssss

Explanation: This message is informational. tt = space type (table space or index space) dd = database ss = space name pp = partition number sss = status value ssss = source or target subsystem name

User response: Determine the cause of the status condition and correct if necessary.

CKZ54721W ALL TARGET CATALOG ACCESSES
COMPLETE, WITH ERRORS, TSOK
aaa, ISOK bbb, TSERR ccc, ISERR ddd

Explanation: This message is informational. aaa = count of table spaces ok bbb = count of indexspaces ok ccc = count of table spaces with errors ddd = count of indexspaces with errors

User response: Determine the cause of the error(s) and resubmit the job.

CKZ54722I ALL TARGET CATALOG ACCESSES
COMPLETED WITHOUT ERRORS,
TABLESPACES aaa, INDEXSPACES bbb

Explanation: This message is informational. aaa = count of table spaces ok bbb = count of indexspaces ok Note that a space that is not found is not counted as an error.

User response: No action is required.

CKZ54723I tt STATUS VALUE sss IS INVALID FOR
DB2 VERSION vvv, DISCARDED

Explanation: This message is informational. tt = status type ADVISORY or RESTRICT sss = status value vvv = DB2 version

User response: No action is required.

CKZ54724W DATASET ddd HAS FAILED STATUS
CHECKING, WILL BE COPIED PER
ALLOW-COPY-ON-MISMATCH(Y) TO
SUBSYSTEM tttt

Explanation: This message is a warning. The copy of this data set will be attempted despite the status failure. For the copy to proceed, MAX_RC must be set to 4. If not, the job will end on the warning. ddd = source data set name tttt = target subsystem

User response: Closely check the viability of the target space with the failed status after the copy completes.

CKZ54725W DATASET dsname, WILL NOT BE
COPIED DUE TO SOURCE AND/OR
TARGET STATUS FAILURE

Explanation: This message is a warning. dsname = data set name

User response: Correct the status failure and resubmit the job.

| **CKZ54726E** Fatal generate DDL error

| **Explanation:** This is an error.

| **User response:** Call IBM Software Support if unable to resolve this error.

| **CKZ54727E** Fatal execute DDL error

| **Explanation:** This is an error.

| **User response:** Call IBM Software Support if unable to resolve this error.

| **CKZ54728I** RESTRICT STATUS added per
| QUIESCE requirement for log apply,
| status_added

| **Explanation:** If a QUIESCE is requested per log apply
| parm, this status check has been added.

| **User response:** No action is required.

| **CKZ54729W** No QUIESCE to be done for data set
| data_set, disallowed per status value(s),
| failing_status

| **Explanation:** A QUIESCE is requested per log apply
| parm and the status check has failed.
| WARN-IF-SKIP-QUIESCE is YES.

| **User response:** Resolve the status and resubmit the
| job.

| **CKZ54730I** No QUIESCE to be done for data set
| *data_set*, disallowed per status value(s),
| *failing_status*

| **Explanation:** A QUIESCE is requested per log apply
| parm and the status check has failed. In addition,
| WARN-IF-SKIP-QUIESCE is NO.

| **User response:** No action is required.

CKZ54731I OBJECT tt ON TARGET ssss, FOUND
DATASET dsname

Explanation: This message is informational. tt = object
type (TS or IS) ssss = DB2 subsystem

User response: No action is required.

CKZ54732I OBJECT tt ON TARGET ssss, NO
OBJECT AVAILABLE, DB2 V9 AND
ABOVE, INSTANCE DEFAULTING
FOR DATASET dsname

Explanation: This message is informational. The target
data set instance defaults to 1. tt = object type (TS or
IS) ssss = target DB2 subsystem dsname = target data
set name

User response: No action is required.

CKZ54733I OBJECT tt ON TARGET ssss, CANNOT
FIND THE TABLESPACE FOR THE
INDEX, DB2 V9 AND ABOVE,
INSTANCE DEFAULTING FOR
DATASET dsname

Explanation: This message is informational. The target
data set instance defaults to 1. tt = object type (TS or
IS) ssss = target DB2 subsystem

User response: No action is required.

CKZ54736I ttt with Non Zero Version Found, ddd,
sss, ccc, ooo

Explanation: This message is informational. An object
with a non-zero version was found in the catalog. This
message prints when COPY subcommand
WARN-ON-VERSIONS is No. ttt = space type: table
space, table, index ddd = current non-zero version sss =
subsystem ccc = creator of the object vvv = name of the
object

User response: For any table spaces or tables with this
message, ensure a REORG and REPAIR VERSIONS has
been done. For any indexes, ensure a REBUILD has
been done.

CKZ54737W ttt with Non Zero Version Found, ddd,
sss, ccc, ooo

Explanation: This message is a warning. An object
with a non-zero version was found in the catalog. This
message prints when COPY subcommand
WARN-ON-VERSIONS is Yes. Use
WARN-ON-VERSIONS(Y) if you suspect one or more
of the spaces with versions need further utility
processing. Note that not all warning messages indicate
that further utility processing is required. ttt = space
type: table space, table, index ddd = current non-zero
version sss = subsystem ccc = creator of the object vvv
= name of the object

User response: For any table spaces or tables with this
message, ensure a REORG and REPAIR VERSIONS has
been done. For any indexes, ensure a REBUILD has
been done.

CKZ54738I Simple Tablespace Found, sss, ccc, ooo

Explanation: This message is informational. A simple
tablespace was found in the catalog. This message
prints when COPY subcommand WARN-ON-SIMPLE-
TABLESPACE is N. sss = subsystem ccc = creator of the
object ooo = name of the object

User response: When copied the first time, ensure the
target object is accessible.

CKZ54739W Simple Tablespace Found, sss, ccc, ooo

Explanation: This message is a warning. A simple
table space was found in the catalog. This message
prints when COPY subcommand WARN-ON-SIMPLE-
TABLESPACE is Y. sss = subsystem ccc = creator of the
object vvv = name of the object

User response: When copied the first time, ensure the
target object is accessible.

CKZ54740W Source Object ooo ssss.ddd.iii cannot be
matched with Target Object tttt.eee.jjj,
src keys=nnn, trg keys=ppp

Explanation: This message is a warning. The number
of columns differ from source to target index. ooo =
object type (indexspace) ssss = source subsystem ddd =
source database iii = source indexspace tttt = target
subsystem ddd = target database jjj = target indexspace
nnn = number of source keys ppp = number of target
keys

User response: Change the source or target index so
that they have the same keys. Resubmit the source job.

CKZ54742I TARGET SS ssss, tt OBJECT WITH
SOURCE DSN=dsname, TARGET
DSN=dsname CAN BE COPIED AS
BOTH IPREFIX I/J EXIST ON THE
TARGET AND THE PARMS ALLOW

DATASET REPLACEMENT

Explanation: This message is informational. ssss = DB2 subsystem tt = object type (TS or IS)

User response: No action is required.

CKZ54743W TARGET SS ssss, tt OBJECT WITH SOURCE DSN=dsname, TARGET DSN=dsname, BOTH EXIST AND THE PARMS DO NOT ALLOW DATASET REPLACEMENT, THE CKZ 5TH LEVEL QUALIFIER, F0001, WILL BE USED FOR THE NEW TARGET DATASET

Explanation: This message is a warning. ssss = DB2 subsystem tt = object type (TS or IS)

User response: Correct the error and resubmit the job.

CKZ54745I TARGET SS ssss, tt OBJECT WITH SOURCE DSN=dsname AND TARGET DSN=dsname, CAN BE COPIED AS THE PARMS ALLOW DATASET REPLACEMENT

Explanation: This message is informational. ssss = DB2 subsystem tt = object type (TS or IS)

User response: No action is required.

CKZ54746I Source table ccc.ttt, Column lll, FOREIGNKEY override using DATASUBTYPE command, was ooo, now nnn

Explanation: This message is informational. The new FOREIGNKEY value will be used to mask data for this column. ccc = source table creator ttt = source table name lll = source column name ooo = old source FOREIGNKEY value nnn = new source FOREIGNKEY value

User response: No action is required.

CKZ54748I Source table ccc.ttt, Column lll, Target table ddd.uuu, Column mmm, both FOREIGNKEY values blank, use product default, S (SBCS)

Explanation: This message is informational. Both foreign key values are blank. The product default S (SBCS) will be used to mask data for this column. ccc = source table creator ttt = source table name lll = source column name ddd = target table creator uuu = target table name mmm = target column name

User response: No action is required.

CKZ54749I Source table ccc.ttt, Column lll, Target table ddd.uuu, Column mmm, source FOREIGNKEY value blank, use target, vvv

Explanation: This message is informational. The source FOREIGNKEY value is blank. The target FOREIGNKEY value will be used to mask data for this column. ccc = source table creator ttt = source table name lll = source column name ddd = target table creator uuu = target table name mmm = target column name vvv = FOREIGNKEY value used to mask this column

User response: No action is required.

CKZ54751I TARGET SS ssss, TT OBJECT WITH SOURCE DSN=dsname AND TARGET DSN=dsname, CAN BE COPIED, NO TARGET DATASETS EXIST, USING IPREFIX DEFAULT F

Explanation: This message is informational. ssss = DB2 subsystem tt = object type (TS or IS)

User response: No action is required.

CKZ54752W FOR OBJECT tt THE SOURCE AND TARGET DATASET NAMES ARE THE SAME, SRCDS = ssss aaa AND TRGDS = tttt bbb

Explanation: This message is a warning that the source and target data set names are identical. No copy will be attempted. tt = object type (TS or IS) ssss = source DB2 subsystem aaa = source data set name tttt = target DB2 subsystem bbb = target data set name This can occur when the copy is on the same subsystem and object translation is used such that the object selected by Listdef is the target name on an OBJ-XLATE command.

User response: Correct the LISTDEF specification and resubmit the job.

CKZ54756W SOURCE OBJECT tt ssss.aaa.bbb.pp CAN NOT BE MATCHED WITH OBJECT tttt.ccc.ddd, WILL NOT BE COPIED

Explanation: This message is a warning that the source partition number did not match the target partition number. tt = object type (Tablepart or Indexpart) ssss = source DB2 subsystem aaa = source database name bbb = source space name (TS or IS) pp = source partition number tttt = target DB2 subsystem ccc = target database name ddd = target space name (TS or IS)

User response: Check the object for the number of partitions in both objects. If the number of target partitions is greater or equal the number of source partitions, then contact the support center. If not,

exclude the target partition(s) that are greater than the largest source partition.

CKZ54757I Source table ccc.ttt, Column lll, Target table ddd.uuu, Column mmm, Target FOREIGNKEY value blank, use Source, vvv

Explanation: This message is informational. The target FOREIGNKEY value is blank. The source FOREIGNKEY value will be used to mask data for this column. ccc = source table creator ttt = source table name lll = source column name ddd = target table creator uuu = target table name mmm = target column name vvv = FOREIGNKEY value used to mask this column

User response: No action is required.

CKZ54758W Source table ccc.ttt, Column lll, FOREIGNKEY vvv, Target table ddd.uuu, Column mmm, FOREIGNKEY www, Not Compatible

Explanation: This message is a warning that the source and target columns may not be compatible.ccc = source table creator ttt = source table name lll = source column name vvv = source FOREIGNKEY value ddd = target table creator uuu = target table name mmm = target column name www = target FOREIGNKEY value

User response: Verify that the source and target values are compatible. If not, ALTER one or both columns as appropriate.

CKZ54759W Dataset Mismatch, Attribute data_set_attribute, Source dsname = YES | NO Target dsname = YES | NO

Explanation: This message is a warning. The data set attribute listed in the message does not match between source and target. There is a high probability that the source data set cannot be copied to the target.

User response: Redefine one or both data sets and resubmit the job.

CKZ54760I object_type mismatch, attribute object_attribute_compared, source_subsystem.source_object_qualifier_1 . source_object_qualifier_2 . halfword_source_value = source_attribute_decimal_value and target_subsystem . target_object_qualifier_1 . target_object_qualifier_2 . halfword_target_value = target_attribute_decimal_value

Explanation: This column's type was changed from LONGVAR to VARCHAR. No incompatibility is reported.

User response: No action is required.

CKZ54762E object_type mismatch, attribute object_attribute_compared, source_DB2_ssid.first_source_object_qualifier. second_source_object_qualifier = source_attribute_decimal_value and target_DB2_ssid.first_target_object_qualifier. second_target_object_qualifier = target_attribute_decimal_value

Explanation: This message is an error. An object attribute does not match between the source and target table space or index space. The OBJECT-MISMATCH-RETURN-CODE command was included and specifies a return code override of 8 for this attribute. No copies are performed, regardless of the setting for ALLOW-COPY-ON-MISMATCH. The details of the object, the attribute that was compared, and the source and target values are listed in the message.

User response: Correct the object and resubmit the job.

CKZ54762I object_type mismatch, attribute object_attribute_compared, source_DB2_ssid.first_source_object_qualifier. second_source_object_qualifier = source_attribute_decimal_value and target_DB2_ssid.first_target_object_qualifier. second_target_object_qualifier = target_attribute_decimal_value

Explanation: This message is informational. An object attribute does not match between the source and target table space or index space. The OBJECT-MISMATCH-RETURN-CODE command was included and specifies a return code override of 0 for this attribute. The details of the object, the attribute that was compared, and the source and target values are listed in the message.

User response: No action is required.

CKZ54762W object_type mismatch, attribute object_attribute_compared, source_DB2_ssid.first_source_object_qualifier. second_source_object_qualifier = source_attribute_decimal_value and target_DB2_ssid.first_target_object_qualifier. second_target_object_qualifier = target_attribute_decimal_value

Explanation: This message is a warning. An object attribute does not match between the source and target table space or index space. This message is the default message if the OBJECT-MISMATCH-RETURN-CODE command was not specified, or if OBJECT-MISMATCH-RETURN-CODE was specified with a return code of 4. The details of the object, the attribute compared, and the source and target values are listed in the message.

User response: Correct the object and resubmit the job.

CKZ54763E *object_type mismatch, attribute*
object_attribute_compared,
source_DB2_ssid.first_source_object_qualifier.
second_source_object_qualifier =
source_attribute_halfword_decimal_value and
target_DB2_ssid.first_target_object_qualifier.
second_target_object_qualifier =
target_attribute_halfword_decimal_value

Explanation: This message is an error. An object attribute does not match between the source and target table space or index space. The OBJECT-MISMATCH-RETURN-CODE command was included and specifies a return code override of 8 for this attribute. No copies are performed, regardless of the setting for ALLOW-COPY-ON-MISMATCH. The details of the object, the attribute that was compared, and the source and target values are listed in the message.

User response: Correct the object and resubmit the job.

CKZ54763I *object_type mismatch, attribute*
object_attribute_compared,
source_DB2_ssid.first_source_object_qualifier.
second_source_object_qualifier =
source_attribute_halfword_decimal_value and
target_DB2_ssid.first_target_object_qualifier.
second_target_object_qualifier =
target_attribute_halfword_decimal_value

Explanation: This message is informational. An object attribute does not match between the source and target table space or index space. The OBJECT-MISMATCH-RETURN-CODE command was included and specifies a return code override of 0 for this attribute. The details of the object, the attribute that was compared, and the source and target values are listed in the message.

User response: No action is required.

CKZ54763W *object_type mismatch, attribute*
object_attribute_compared,
source_DB2_ssid.first_source_object_qualifier.
second_source_object_qualifier =
source_attribute_halfword_decimal_value and
target_DB2_ssid.first_target_object_qualifier.
second_target_object_qualifier =
target_attribute_halfword_decimal_value

Explanation: This message is a warning. An object attribute does not match between the source and target table space or index space. This message is the default message if the OBJECT-MISMATCH-RETURN-CODE command was not specified, or if OBJECT-MISMATCH-RETURN-CODE was specified with a return code of 4. The details of the object, the attribute compared, and the source and target values are listed in the message.

User response: Correct the object and resubmit the job.

CKZ54764I *ooo MISMATCH, ATTRIBUTE aaa,*
ssss.ddd.nnn.ppp = vvv And
tttt.eee.fff.qqq = xxx, CAN BE
CORRECTED WITH COPY

Explanation: This message is a warning. ooo = object type aaa = object attribute compared ssss = source DB2 subsystem ddd = first level source object qualifier nnn = second level source object qualifier ppp = source partition number vvv = source attribute fullword decimal value tttt = target DB2 subsystem eee = first level target object qualifier fff = second level target object qualifier qqq = target partition number xxx = target attribute fullword decimal value

User response: One of the DB2 subsystems has a table space or index space that is not defined as page size = CISIZE. Consider changing the CISIZE to match the page size for DB2 performance.

CKZ54765W **DDL processing and one or more objects are incompatible.**

Explanation: This message is a warning.

User response: Resolve the incompatibilities and rerun the job.

CKZ54767E *object_type mismatch, attribute*
object_attribute_compared,
source_DB2_ssid.first_source_object_qualifier.
second_source_object_qualifier =
source_attribute_value and
target_DB2_ssid.first_target_object_qualifier.
second_target_object_qualifier =
target_attribute_value

Explanation: This message is an error. An object attribute does not match between the source and target table space or index space. The OBJECT-MISMATCH-RETURN-CODE command was included and specifies a return code override of 8 for this attribute. No copies are performed, regardless of the setting for ALLOW-COPY-ON-MISMATCH. The details of the object, the attribute that was compared, and the source and target values are listed in the message.

User response: Correct the object and resubmit the job.

CKZ54767I *object_type mismatch, attribute*
object_attribute_compared,
source_DB2_ssid.first_source_object_qualifier.
second_source_object_qualifier =
source_attribute_value and
target_DB2_ssid.first_target_object_qualifier.
second_target_object_qualifier =
target_attribute_value

Explanation: This message is informational. An object attribute does not match between the source and target table space or index space. The OBJECT-MISMATCH-

| RETURN-CODE command was included and specifies
 | a return code override of 0 for this attribute. The
 | details of the object, the attribute that was compared,
 | and the source and target values are listed in the
 | message.

| **User response:** No action is required.

| **CKZ54767W** *object_type mismatch, attribute*
 | *object_attribute_compared,*
 | *source_DB2_ssid.first_source_object_qualifier.*
 | *second_source_object_qualifier =*
 | *source_attribute_value and*
 | *target_DB2_ssid.first_target_object_qualifier.*
 | *second_target_object_qualifier =*
 | *target_attribute_value*

| **Explanation:** This message is a warning. An object
 | attribute does not match between the source and target
 | table space or index space. This message is the default
 | message if the OBJECT-MISMATCH-RETURN-CODE
 | command was not specified, or if OBJECT-MISMATCH-
 | RETURN-CODE was specified with a return code of 4.
 | The details of the object, the attribute compared, and
 | the source and target values are listed in the message.

| **User response:** Correct the object and resubmit the
 | job.

| **CKZ54768I** **Row Format Incompatibility Bypassed**
 | **per the IGNORE-RF-MISMATCH-IF-**
 | **NO-VAR-COLS(Y) Command,**
 | **DB=source-database, TS=source-table-space**

| **Explanation:** This message is informational. It will
 | print once for each partition if the tablespace is
 | partitioned.

| **User response:** No action is required.

| **CKZ54769E** *object_type mismatch, attribute*
 | *object_attribute_compared,*
 | *source_DB2_ssid.first_source_object_qualifier.*
 | *second_source_object_qualifier.*
 | *halfword_source_value =*
 | *source_attribute_value and target_DB2_ssid.*
 | *first_target_object_qualifier.*
 | *second_target_object_qualifier.*
 | *halfword_target_value =*
 | *target_attribute_value*

| **Explanation:** This message is an error. An object
 | attribute does not match between the source and target
 | table space or index space. The OBJECT-MISMATCH-
 | RETURN-CODE command was included and specifies
 | a return code override of 8 for this attribute. No copies
 | are performed, regardless of the setting for
 | ALLOW-COPY-ON-MISMATCH. The details of the
 | object, the attribute that was compared, and the source
 | and target values are listed in the message.

| **User response:** Correct the object and resubmit the
 | job.

| **CKZ54769I** *object_type mismatch, attribute*
 | *object_attribute_compared,*
 | *source_DB2_ssid.first_source_object_qualifier.*
 | *second_source_object_qualifier.*
 | *halfword_source_value =*
 | *source_attribute_value and target_DB2_ssid.*
 | *first_target_object_qualifier.*
 | *second_target_object_qualifier.*
 | *halfword_target_value =*
 | *target_attribute_value*

| **Explanation:** This message is informational. An object
 | attribute does not match between the source and target
 | table space or index space. The OBJECT-MISMATCH-
 | RETURN-CODE command was included and specifies
 | a return code override of 0 for this attribute. The
 | details of the object, the attribute that was compared,
 | and the source and target values are listed in the
 | message.

| **User response:** No action is required.

| **CKZ54769W** *object_type mismatch, attribute*
 | *object_attribute_compared,*
 | *source_DB2_ssid.first_source_object_qualifier.*
 | *second_source_object_qualifier.*
 | *halfword_source_value =*
 | *source_attribute_value and target_DB2_ssid.*
 | *first_target_object_qualifier.*
 | *second_target_object_qualifier.*
 | *halfword_target_value =*
 | *target_attribute_value*

| **Explanation:** This message is a warning. An object
 | attribute does not match between the source and target
 | table space or index space. This message is the default
 | message if the OBJECT-MISMATCH-RETURN-CODE
 | command was not specified, or if OBJECT-MISMATCH-
 | RETURN-CODE was specified with a return code of 4.
 | The details of the object, the attribute that was
 | compared, and the source and target values are listed
 | in the message.

| **User response:** Correct the object and resubmit the
 | job.

| **CKZ54770E** **Subtask** *subtask_number*, **IDCAMS**
 | **Execute Error, RC=return_code,**
 | **RS=IDCAMS_reason_code, Text to**
 | **Follow...**

| **Explanation:** This message is an internal error.
 | Additional text may follow this message.

| **User response:** Contact IBM Software Support.

| **CKZ54771I** **Subtask** *subtask_number*, **IDCAMS**
 | **Messages Follow...**

| **Explanation:** This message is informational.

| **User response:** No action is required.

CKZ54772E *object_type mismatch, attribute object_attribute_compared, source_DB2_ssid.first_source_object_qualifier. second_source_object_qualifier. source_value = halfword_source_attribute_value and target_DB2_ssid.first_target_object_qualifier. second_target_object_qualifier. target_value = halfword_target_attribute_value*

Explanation: This message is an error. An object attribute does not match between the source and target table space or index space. The OBJECT-MISMATCH-RETURN-CODE command was included and specifies a return code override of 8 for this attribute. No copies are performed, regardless of the setting for ALLOW-COPY-ON-MISMATCH. The details of the object, the attribute that was compared, and the source and target values are listed in the message.

User response: Correct the object and resubmit the job.

CKZ54772I *object_type mismatch, attribute object_attribute_compared, source_DB2_ssid.first_source_object_qualifier. second_source_object_qualifier. source_value = halfword_source_attribute_value and target_DB2_ssid.first_target_object_qualifier. second_target_object_qualifier. target_value = halfword_target_attribute_value*

Explanation: This message is informational. An object attribute does not match between the source and target table space or index space. The OBJECT-MISMATCH-RETURN-CODE command was included and specifies a return code override of 0 for this attribute. The details of the object, the attribute that was compared, and the source and target values are listed in the message.

User response: No action is required.

CKZ54772W *object_type mismatch, attribute object_attribute_compared, source_DB2_ssid.first_source_object_qualifier. second_source_object_qualifier. source_value = halfword_source_attribute_value and target_DB2_ssid.first_target_object_qualifier. second_target_object_qualifier. target_value = halfword_target_attribute_value*

Explanation: This message is a warning. An object attribute does not match between the source and target table space or index space. This message is the default message if the OBJECT-MISMATCH-RETURN-CODE command was not specified, or if OBJECT-MISMATCH-RETURN-CODE was specified with a return code of 4. The details of the object, the attribute that was compared, and the source and target values are listed in the message.

User response: Correct the object and resubmit the job.

CKZ54773E *object_type mismatch, attribute object_attribute_compared, source_DB2_ssid.first_source_object_qualifier. second_source_object_qualifier. source_value = source_attribute_value and target_DB2_ssid.first_target_object_qualifier. second_target_object_qualifier. target_value = target_attribute_value*

Explanation: This message is an error. An object attribute does not match between the source and target table space or index space. The OBJECT-MISMATCH-RETURN-CODE command was included and specifies a return code override of 8 for this attribute. No copies are performed, regardless of the setting for ALLOW-COPY-ON-MISMATCH. The details of the object, the attribute that was compared, and the source and target values are listed in the message.

User response: Correct the object and resubmit the job.

CKZ54773I *object_type mismatch, attribute object_attribute_compared, source_DB2_ssid.first_source_object_qualifier. second_source_object_qualifier. source_value = source_attribute_value and target_DB2_ssid.first_target_object_qualifier. second_target_object_qualifier. target_value = target_attribute_value*

Explanation: This message is informational. An object attribute does not match between the source and target table space or index space. The OBJECT-MISMATCH-RETURN-CODE command was included and specifies a return code override of 0 for this attribute. The details of the object, the attribute that was compared, and the source and target values are listed in the message.

User response: No action is required.

CKZ54773W *object_type mismatch, attribute object_attribute_compared, source_DB2_ssid.first_source_object_qualifier. second_source_object_qualifier. source_value = source_attribute_value and target_DB2_ssid.first_target_object_qualifier. second_target_object_qualifier. target_value = target_attribute_value*

Explanation: This message is a warning. An object attribute does not match between the source and target table space or index space. This message is the default message if the OBJECT-MISMATCH-RETURN-CODE command was not specified, or if OBJECT-MISMATCH-RETURN-CODE was specified with a return code of 4. The details of the object, the attribute that was

compared, and the source and target values are listed in the message.

User response: Correct the object and resubmit the job.

CKZ54777I ooo MISMATCH, ATTRIBUTE aaa, ssss.ddd.nnn.ppp = vvv AND tttt.eee.fff.qqq = xxx, yyy

Explanation: This message is a warning. ooo = object type aaa = object attribute compared ssss = source DB2 subsystem ddd = first level source object qualifier nnn = second level source object qualifier ppp = source value vvv = source attribute value tttt = target DB2 subsystem eee = first level target object qualifier fff = second level target object qualifier qqq = target value xxx = target attribute value yyy = (optional) can be Corrected with ALTER TABLE(DB2 Version 8 and higher)

User response: Correct the object(s) and resubmit the job. If yyy is present, you must have the SQL OUT DD in the source and target jobs.

CKZ54778E object_type mismatch, attribute object_attribute_compared, source_DB2_ssid.first_source_object_qualifier.second_source_object_qualifier.source_value = decimal_source_attribute_value and target_DB2_ssid.first_target_object_qualifier.second_target_object_qualifier.target_value = decimal_target_attribute_value

Explanation: This message is an error. An object attribute does not match between the source and target table space or index space. The OBJECT-MISMATCH-RETURN-CODE command was included and specifies a return code override of 8 for this attribute. No copies are performed, regardless of the setting for ALLOW-COPY-ON-MISMATCH. The details of the object, the attribute that was compared, and the source and target values are listed in the message.

User response: Correct the object and resubmit the job.

CKZ54778I object_type mismatch, attribute object_attribute_compared, source_DB2_ssid.first_source_object_qualifier.second_source_object_qualifier.source_value = decimal_source_attribute_value and target_DB2_ssid.first_target_object_qualifier.second_target_object_qualifier.target_value = decimal_target_attribute_value

Explanation: This message is informational. An object attribute does not match between the source and target table space or index space. The OBJECT-MISMATCH-RETURN-CODE command was included and specifies a return code override of 0 for this attribute. The details of the object, the attribute that was compared,

and the source and target values are listed in the message.

User response: No action is required.

CKZ54778W object_type mismatch, attribute object_attribute_compared, source_DB2_ssid.first_source_object_qualifier.second_source_object_qualifier.source_value = decimal_source_attribute_value and target_DB2_ssid.first_target_object_qualifier.second_target_object_qualifier.target_value = decimal_target_attribute_value

Explanation: This message is a warning. An object attribute does not match between the source and target table space or index space. This message is the default message if the OBJECT-MISMATCH-RETURN-CODE command was not specified, or if OBJECT-MISMATCH-RETURN-CODE was specified with a return code of 4. The details of the object, the attribute that was compared, and the source and target values are listed in the message.

User response: Correct the object and resubmit the job.

CKZ54779W MISMATCH FOR INDEX ssss.ddd.nnn AND tttt.eee.fff DUE TO MISMATCH FOR TABLE ssss.hhh.iii AND tttt.iii.mmm

Explanation: This message is a warning. ssss = source DB2 subsystem ddd = source index creator nnn = source index name tttt = target DB2 subsystem eee = target index creator fff = target index name hhh = source table creator iii = source table name lll = target table creator mmm = target table name

User response: Correct the table object(s) and resubmit the job.

CKZ54780I Target SSID target_subsystem_ID, object in DB2 catalog, space_type dsname. No data sets found; will be copied using target IJ indicator per the COPY-IJ-TO-NONEXISTENT-TARGET command

Explanation: This message is informational. The target data sets will be created using the target IJ fifth node indicator.

User response: No response is required.

CKZ54781I Target SSID target_subsystem_ID, object not in DB2 catalog, space_type dsname. No data sets found; will be copied using source IJ indicator per the COPY-IJ-TO-NONEXISTENT-TARGET command

Explanation: This message is informational. The target

data sets are not in the target catalog, therefore the target data sets will be created using the source IJ fifth node indicator.

User response: No response is required.

CKZ54782E *object_type mismatch, attribute compared_attribute, sourceDB2.first_source_object_qualifier. second_source_object_qualifier = source_attribute_value and targetDB2.first_target_object_qualifier. second_target_object_qualifier = target_attribute_value*

Explanation: This message is an error. An object attribute does not match between the source and target table space or index space. The OBJECT-MISMATCH-RETURN-CODE command was included and specifies a return code override of 8 for this attribute. No copies are performed, regardless of the setting for ALLOW-COPY-ON-MISMATCH. The details of the object, the attribute that was compared, and the source and target values are listed in the message.

User response: Correct the object and resubmit the job.

CKZ54782I *object_type mismatch, attribute compared_attribute, sourceDB2.first_source_object_qualifier. second_source_object_qualifier = source_attribute_value and targetDB2.first_target_object_qualifier. second_target_object_qualifier = target_attribute_value*

Explanation: This message is informational. An object attribute does not match between the source and target table space or index space. The OBJECT-MISMATCH-RETURN-CODE command was included and specifies a return code override of 0 for this attribute. The details of the object, the attribute that was compared, and the source and target values are listed in the message.

User response: No action is required.

CKZ54782W *object_type mismatch, attribute compared_attribute, sourceDB2.first_source_object_qualifier. second_source_object_qualifier = source_attribute_value and targetDB2.first_target_object_qualifier. second_target_object_qualifier = target_attribute_value*

Explanation: This message is a warning. An object attribute does not match between the source and target table space or index space. This message is the default message if the OBJECT-MISMATCH-RETURN-CODE command was not specified, or if OBJECT-MISMATCH-

RETURN-CODE was specified with a return code of 4. The details of the object, the attribute that was compared, and the source and target values are listed in the message.

User response: Correct the object and resubmit the job.

CKZ54783E *Data set data_set_name has failed object match checking with a return code of 8 and will not be copied to subsystem target_subsystem_ID*

Explanation: The data set that is listed in the message failed object mismatch checking and cannot be copied.

User response: Correct the object and resubmit the job.

CKZ54786E *object_type mismatch, attribute object_attribute_compared, source_DB2_ssid.first_source_object_qualifier. second_source_object_qualifier. halfword_source_value = source_attribute_value (in hexadecimal) and target_DB2_ssid.first_target_object_qualifier. second_target_object_qualifier. halfword_target_value = target_attribute_value (in hexadecimal)*

Explanation: This message is an error. An object attribute does not match between the source and target table space or index space. The OBJECT-MISMATCH-RETURN-CODE command was included and specifies a return code override of 8 for this attribute. No copies are performed, regardless of the setting for ALLOW-COPY-ON-MISMATCH. The details of the object, the attribute that was compared, and the source and target values are listed in the message.

User response: Correct the object and resubmit the job.

CKZ54786I *object_type mismatch, attribute object_attribute_compared, source_DB2_ssid.first_source_object_qualifier. second_source_object_qualifier. halfword_source_value = source_attribute_value (in hexadecimal) and target_DB2_ssid.first_target_object_qualifier. second_target_object_qualifier. halfword_target_value = target_attribute_value (in hexadecimal)*

Explanation: This message is informational. An object attribute does not match between the source and target table space or index space. The OBJECT-MISMATCH-RETURN-CODE command was included and specifies a return code override of 0 for this attribute. The details of the object, the attribute that was compared, and the source and target values are listed in the message.

| **User response:** No action is required.

| **CKZ54786W** *object_type mismatch, attribute*
 | *object_attribute_compared,*
 | *source_DB2_ssid.first_source_object_qualifier.*
 | *second_source_object_qualifier.*
 | *halfword_source_value =*
 | *source_attribute_value (in hexadecimal) and*
 | *target_DB2_ssid.first_target_object_qualifier.*
 | *second_target_object_qualifier.*
 | *halfword_target_value =*
 | *target_attribute_value (in hexadecimal)*

| **Explanation:** This message is a warning. An object attribute does not match between the source and target table space or index space. This message is the default message if the OBJECT-MISMATCH-RETURN-CODE command was not specified, or if OBJECT-MISMATCH-RETURN-CODE was specified with a return code of 4. The details of the object, the attribute that was compared, and the source and target values are listed in the message.

| **User response:** Correct the object and resubmit the job.

CKZ54787I **Data set extension mismatch - source**
 DSN=*data_set_name*,
num_of_source_extensions **data sets, target**
 DSN=*data_set_name*,
num_of_target_extensions **data sets.**

Explanation: The number of extensions for the source and target data sets do not match, and WARN-ON-DATASET-EXTENSION-MISMATCH is set to N; therefore, this message is informational only.

User response: No action is required.

| **CKZ54788W** **Data set extension mismatch - source**
 | DSN=*data_set_name*,
 | *num_of_source_extensions* **data sets, target**
 | DSN=*data_set_name*,
 | *num_of_target_extensions* **data sets.**

| **Explanation:** The number of extensions for the source and target data sets do not match, and WARN-ON-DATASET-EXTENSION-MISMATCH is set to Y; therefore, this message is a warning. The job return code is set to 4.

| **User response:** Determine the reason for the extension mismatch, correct the objects, and resubmit the job.

CKZ54789W **DATASET dsname, WILL NOT BE COPIED DUE TO SOURCE AND TARGET OBJECT MISMATCH**

Explanation: This message is a warning.

User response: Correct the object mismatch and resubmit the job.

CKZ54792E **REMOTE CONNECT NOT COMPLETED, ccc**

Explanation: This message is either a user error or a system error. Unable to connect to the target subsystem. ccc = error type

User response: If the error type is UNKNOWN, contact IBM Software Support. If the error type is ERROR, determine the cause of the error(s) and resubmit the job. This could be a parameter error. If not, contact the systems programmer or data base administrator.

CKZ54793I **REMOTE CONNECT COMPLETED, USING sss**

Explanation: This message is informational. It may be followed by a failure at the target if the userid or password are invalid. This could be indicated by SQLCODE = -30082. sss = connect type (CAF, DDF or TCP/IP)

User response: No action is required.

CKZ54801I **FUZZYCPY(Y) SPECIFIED, NO SOURCE SPACES WILL BE STOPPED. USER MUST VALIDATE TARGET DATASET(S)**

Explanation: This message is informational. It is output when COPY parameter FUZZY-COPY is Y.

User response: No action is required.

CKZ54802W **ALL COPY REQUESTS COMPLETED WITH ERRORS, TSOK aaa, ISOK bbb, TSERR ccc, ISERR ddd**

Explanation: This message is informational. aaa = count of table spaces ok bbb = count of indexspaces ok ccc = count of table spaces with errors ddd = count of indexspaces with errors. If MAX_RC=4, one of more data sets may have been excluded from the copy due to a warning. These are not included in these statistics.

User response: Determine the cause of the error(s) and resubmit the job.

CKZ54803I **ALL ATTEMPTED sss REQUESTS COMPLETED WITHOUT ERRORS, TABLESPACES aaa, INDEXSPACES bbb**

Explanation: This message is informational. sss = COPY or Simulate aaa = count of table spaces ok bbb = count of indexspaces ok

User response: No action is required.

CKZ54804W COPY aborted, one or more copies may not complete.

Explanation: This message is a warning. Copies may have been aborted due to a previous error.

User response: Review other error messages. If unable to resolve errors, contact IBM Software Support.

CKZ54821I - BEGIN SIMULATE(A) STATUS REPORT

Explanation: This message is informational. It begins a report of data sets processed. Report fields follow... RC = processing return code SPACE TYPE = TS table space IS indexspace LS LOB space CLONE = blank if not cloned Y is a cloned space (V9 and higher) B is a base space (V9 and higher) REPL DSN = replace target data set with this copy Y target does exist N target does not exist TRG OBJ = target object exists Y target does exist N target does not exist OBJ XLATE DB database name translation result TS table space name translation result IS indexspace name translation result IXC index creator translation result IXN index name translation result blank = no translation - = translation match but not used + = translation match and used

User response: No action is required.

CKZ54822I - END SIMULATE(A) STATUS REPORT

Explanation: This message is informational. It ends the report.

User response: No action is required.

CKZ54823I - BEGIN COPY PGM(NONE) STATUS REPORT

Explanation: This message is informational. It begins a report of data sets processed. Report fields follow:

- RC = processing return code
- SPACE TYPE = TS (table space); IS (indexspace); LS (LOB space)
- CLONE = blank if not cloned; Y is a cloned space (V9 and higher); B is a base space (V9 and higher)
- REPL DSN = replace target data set with this copy; Y = target does exist, N = target does not exist
- TRG OBJ = target object exists; Y = target does exist; N = target does not exist
- OBJ XLATE = DB (database name translation result); TS (table space name translation result); IS (indexspace name translation result); IXC (index creator translation result); IXN (index name translation result) blank = no translation; - = translation match but not used; += translation match and used

User response: No action is required.

CKZ54824I - END COPY PGM(NONE) STATUS REPORT

Explanation: This message is informational. It ends the report.

User response: No action is required.

CKZ54825I - BEGIN COPY COMPLETION STATUS REPORT

Explanation: This message is informational. It begins a report of data sets processed. Report fields follow... RC = processing return code n/a = not attempted or unable to find a copy return code SPACE TYPE = TS table space IS indexspace LS LOB space CLONE = blank if not cloned Y is a cloned space (V9 and higher) B is a base space (V9 and higher) REPL DSN = replace target data set with this copy Y target does exist N target does not exist TRG OBJ = target object exists Y target does exist N target does not exist OBJ XLATE DB database name translation result TS table space name translation result IS indexspace name translation result IXC index creator translation result IXN index name translation result blank = no translation - = translation match but not used + = translation match and used

User response: No action is required.

CKZ54826I - END COPY COMPLETION STATUS REPORT

Explanation: This message is informational. It ends the report.

User response: No action is required.

**CKZ54901E Unable to load program:
load_module_name**

Explanation: A log apply module could not be loaded.

User response: Call IBM Software Support if unable to resolve this error.

**CKZ54902E Bad return from load_module_name,
RC=return_code, RS=reason_code**

Explanation: A log apply module could not be loaded.

User response: Call IBM Software Support if unable to resolve this error.

**CKZ54903E Bad eyecatcher, B call,
first_8_characters_of_eyecatcher**

Explanation: This is an internal error.

User response: Call IBM Software Support.

CKZ54904E Log apply, unable to match minilog data set with log apply control blocks

Explanation: This is probably an internal error.

User response: Call IBM Software Support.

CKZ54905E Log apply, unable to match space database.table_space_or_indexspace.partition Control Blocks

Explanation: This is probably an internal error.

User response: Call IBM Software Support.

CKZ54906I Call DB2 source subsystem, waiting...

Explanation: This message is informational. This DB2 call may take several minutes to complete.

User response: No action is required.

CKZ54907I Number of SPACES-PER-MINILOG changed, now new_number_of_spaces_per_minilog

Explanation: This message is informational. Only 36 minilog data sets can be allocated. This caused the number of SPACES-PER-MINILOG to be changed from that requested in the LOG-APPLY parameters.

User response: No action is required.

CKZ54909E Log apply init call failed, RC=return_code, RS=reason_code

Explanation: A severe error occurred when log apply was processing a source server request.

User response: If unable to resolve this error, contact IBM Software Support.

CKZ54910W Log apply init call completed with warning(s) RC=return_code, RS=reason_code

Explanation: An error occurred when log apply was processing a source server request. It is possible that the error was generated from a single data set.

User response: Check the messages that precede this message to determine which objects have an associated warning. The return code field in the source report cannot be used to determine which objects have a warning. Call IBM Software Support if unable to resolve this error.

CKZ54990I Log apply initialization successful

Explanation: Log apply is ready to begin processing log records.

User response: No action is required.

CKZ54991E Log apply initialization failure, RC=return_code

Explanation: Log apply has failed to initialize. Check for log apply error messages preceding this message. See CKZ54999I for message output.

User response: Call IBM Software Support if unable to resolve this error.

CKZ54992W Log apply initialization warning, RC=return_code

Explanation: Log apply has initialized with one or more warnings.

User response: Check the messages that precede this error message; refer to CKZ54999I for message output. Call IBM Software Support if unable to resolve this error.

CKZ54999I CKZG-B, message_text

Explanation: A message from log apply initialization call. It could be informational, a warning or an error.

User response: No action is required, unless the call ends in an error.

CKZ55001E QUIESCE error, RC=return_code RS=reason_code

Explanation: The QUIESCE utility returned an error.

User response: Call IBM Software Support if unable to resolve this error.

CKZ55002I data_set_type, total DSNs to process = number_of_data_sets, DSNs per utility call = number_of_data_sets_processed_per_call

Explanation: The QUIESCE utility is split into multiple calls per the UTILITY-COMMAND-EXECUTE-PERCENT SET command. *data_set_type* can be BaseTbl (base table) or CloneTbl (clone table).

User response: No action is required.

CKZ55101I Repository use in effect

Explanation: Both DD cards associated with the repository, CKZRRJOB and CKZRRDSN are present and the SET command USE-RUNTIME-REPOSITORY(Y) is present.

User response: No action is required.

CKZ55102E Repository use error, one or both DD cards missing

Explanation: This message is output only if the SET command USE-RUNTIME-REPOSITORY(Y) is present. It is an error.

| **User response:** Ensure both DD cards are available in the JCL stream or turn USE-RUNTIME-REPOSITORY off (N).

| **CKZ55103E** Subtask *subtask_number*, VSAM GET error, R15=*vsam_return_code*, RPLFDBK=*vsam_feedback_code*, DD=*ddname*

| **Explanation:** This message is an error. Unable to access the runtime repository data set.

| **User response:** Check the return and feedback codes to determine the problem. For information about the codes, refer to DFSMS Macro Instructions for Data Sets for your version of z/OS. Call IBM Software Support if unable to resolve this error.

| **CKZ55104I** VSAM repository data set open, DD is *ddname*

| **Explanation:** This message is informational.

| **User response:** No action is required.

| **CKZ55105E** Subtask *subtask_number*, VSAM PUT error, R15=*vsam_return_code*, RPLFDBK=*vsam_feedback_code*, DD=*ddname*

| **Explanation:** This message is an error. Unable to access the runtime repository dataset.

| **User response:** Check the return and feedback codes to determine the problem. For information about the codes, refer to DFSMS Macro Instructions for Data Sets for your version of z/OS. Call IBM Software Support if unable to resolve this error.

| **CKZ55106E** Subtask *subtask_number*, VSAM POINT error, R15=*vsam_return_code*, RPLFDBK=*vsam_feedback_code*, DD=*ddname*

| **Explanation:** This message is an error. Unable to access the runtime repository dataset.

| **User response:** Check the return and feedback codes to determine the problem. For information about the codes, refer to DFSMS Macro Instructions for Data Sets for your version of z/OS. Call IBM Software Support if unable to resolve this error.

| **CKZ55107I** VSAM repository data set close, DD is *ddname*

| **Explanation:** This message is informational.

| **User response:** No action is required.

| **CKZ55200E** No repository DDs found

| **Explanation:** This message is an error. Neither of the input DDs were found, CKZRRJOB or CKZRRDSN.

| **User response:** Correct the error and resubmit the job.

| **CKZ55201E** No report output DDs found

| **Explanation:** This message is an error. None of the four output DD cards were found: CKZJREPL, CKZJREPS, CKZDREPL, CKZDREPS.

| **User response:** Correct the error and resubmit the job.

| **CKZ55202W** *report_type* | *job* | *data_set* report requested, but no *ddname* DD found

| **Explanation:** This message is an error. The report requested requires a DD card for input that was not present.

| **User response:** Correct the error and resubmit the job.

| **CKZ55401W** Source *table_space_type* table space *database.table_space* is not creating log records

| **Explanation:** The table space listed in the message is defined LOGGED NO, so there are no log records to apply. This message only applies when PGM(SRCIMCPY) and is a warning.

| **User response:** Remove the table space from the LISTDEF, change the definition of the table space to enable logging, or change WARN-IF-TS-DEFINED-LOG-NO(Y) to (N) if logging is not required for this table space. Rerun the job.

| **CKZ55402I** Source *table_space_type* table space *database.table_space* is not creating log records

| **Explanation:** The table space listed in the message is defined LOGGED NO, so there are no log records to apply. This message only applies when PGM(SRCIMCPY) and is informational.

| **User response:** No action is required.

| **CKZ55501E** REBUILD error, RC=*return_code* RS=*reason_code*

| **Explanation:** The REBUILD utility returned an error.

| **User response:** Call IBM Software Support if unable to resolve this error.

CKZ56201I TRACING BUFFER WRAP, POSSIBLE LOOP

Explanation: This is an informational WTO. This will only occur if support has requested that tracing be turned on to investigate a problem.

User response: Increase the size of the tracing buffer and resubmit the job.

CKZ56201E Subtask nn, IGGCSI00 Error Return, R15=ggg CSI Module ID=mmm, CSI Return Code is rrr, CSI Reason Code is sss

Explanation: This message is an error. nn = subtask number ggg = R15 return code mmm = 2 character module id from SMS rrr = SMS return code sss = SMS reason code

User response: Call IBM Software Support if the data set exists and is cataloged.

CKZ56202E SUBTASK, CREATION FAILURE, RC=rrr

Explanation: This is probably an internal error. rrr = return code

User response: Call IBM Software Support.

CKZ56203E SUBTASK nn, IGGCSI00 ERROR RETURN, R15=ggg CSI MODULE ID=mmm, CSI RETURN CODE IS rrr, CSI REASON CODE IS sss, DSN=ddd

Explanation: This is probably a catalog error. nn = subtask number ggg = R15 return code mmm = Two character module id from SMS rrr = SMS return code sss = SMS reason code ddd = data set name requested

User response: See the DFSMS Managing Catalogs publication for your version of z/OS for more information on IGGCSI00. Call IBM Software Support if unable to resolve this error.

CKZ56204E SUBTASK nn, IGGCSI00 RETURN WORK AREA ERROR, ttt, hh1, hh2, hh3, DSN=ddd, CATALOG=ccc

Explanation: This is probably an internal error. nn = subtask number (or blank) ttt = error type hh1 = hex value 1 hh2 = hex value 1 hh3 = hex value 1 ddd = data set name requested ccc = z/OS catalog returned

User response: See the DFSMS Managing Catalogs publication for your version of z/OS for more information on IGGCSI00. Call IBM Software Support.

CKZ56401E SUBTASK nn, COMPLETED WITH ERRORS, RC=rrr, RS=sss

Explanation: This message is an error message. It indicates the subtask has output one or more error messages. Messages output by a subtask will have 'Subtask' followed by a number to identify the issuing subtask. nn = subtask number rrr = return code sss = reason code

User response: Check CKZPRINT for error messages output by the subtask.

CKZ56402W SUBTASK nn, COMPLETED WITH WARNINGS, RC=rrr, RS=sss

Explanation: This message is a warning message. It indicates the subtask has output one or more warning messages. Messages output by a subtask will have 'Subtask' followed by a number to identify the issuing subtask. nn = subtask number rrr = return code sss = reason code

User response: Check CKZPRINT for warning messages output by the subtask.

CKZ56403W SUBTASK nn, COMPLETED WITH WARNING(S)

Explanation: This message is a warning message. It indicates the subtask has output one or more warning messages. Messages output by a subtask will have 'Subtask' followed by a number to identify the issuing subtask. nn = subtask number

User response: Check CKZPRINT for warning messages output by the subtask.

CKZ56404I SUBTASK nn, COMPLETED WITH NO ERRORS

Explanation: This message is informational. nn = subtask number

User response: No action is required.

CKZ56497E ABEND 900,REASON=2

Explanation: This terminates the job with an ABEND.

User response: Call IBM Software Support.

CKZ56498E ABEND 900,REASON=1

Explanation: This terminates the job with an ABEND.

User response: Call IBM Software Support.

**CKZ56499I TRACE PRINT TERMINATED,
WRAPPED**

Explanation: This is an informational WTO. This will only occur if support has requested that tracing be turned on to investigate a problem.

User response: Increase the size of the tracing buffer and resubmit the job.

**CKZ56501E INVALID ERROR ID, MESSAGE
DELETED**

Explanation: This is an internal error.

User response: Call IBM Software Support.

CKZ56597E WTO BUFFER ALLOCATION ERROR

Explanation: This is an error WTO. There is not enough virtual storage.

User response: Increase the region size and resubmit the job.

CKZ56598E QUEUEING ERROR IN QMSG

Explanation: This is an error WTO.

User response: Call IBM Software Support.

CKZ56599E GETMAIN ERROR IN QMSG

Explanation: This is an error WTO. There is not enough virtual storage.

User response: Increase the region size and resubmit the job.

**CKZ56901E INFRA LOAD ERROR, RC=rrr, RS=sss,
MODULE=mmm**

Explanation: This message is an error message. It indicates support processing cannot load a required module. rrr = return code sss = reason code mmm = module name

User response: Check the JCL to ensure the STEPLIB is pointing to the correct library and that the install was done correctly. Resubmit the job when corrections made.

**CKZ56902E MASK CHARACTER ERROR, SRC
MASK = sss, TRG MASK = ttt**

Explanation: This message is an error message. It indicates that the given source target object translation mask pair is invalid. Mask pairs may be invalid for the following reason: %, _%, or %_ are in the source and target masks without an interspersed non-mask character. For example: %%% %a%b% is valid %a%b% %%% is valid ._ .% is valid .% .%.. is an error %_ %_ is an error %_ .%_ is an error sss = source mask

value ttt = target mask value

User response: Correct the mask and resubmit the job.

**CKZ56903E MASK CHARACTERS NOT ALIGNED
CORRECTLY, REASON = rrr, SRC
MASK = sss, TRG MASK = ttt**

Explanation: This message is an error message. It indicates that the given source target object translation mask pair is invalid. rrr = reason code: 71 - premature target mask end 72 - premature source mask end 73 - mask characters not matching from source to target, ordinally 74 - total number of each mask character type does not match sss = source mask value ttt = target mask value

User response: Correct the mask and resubmit the job.

**CKZ56904I OBJECT TRANSLATE tt IN USE,
SRC=sss, TRG=ttt, SRC MASK=mmm,
TRG MASK=nnn**

Explanation: This message is informational. tt = 2 digit object type being translated sss = source object name ttt = target object name mmm = source mask value nnn = target mask value

User response: No action is required.

**CKZ57035E GETHOSTBYNAME FOR DOMAIN
ddd AND HOSTNAME hhh FAILED
LOOKUP**

Explanation: This message is a user error. ddd = TCPIP domain name hhh = TCPIP host name

User response: Specify the correct domain and host names and resubmit the job.

**CKZ57301E SUBTASK nn, DETACH ERROR,
CODE=sss**

Explanation: This is an error message that an MVS task cannot be detached. The return from the detach is in the reason code. nn = subtask number sss = reason code

User response: Call IBM Software Support.

CKZ57302I ALL SUBTASKS NOW DETACHED

Explanation: This message is informational.

User response: No action is required.

CKZ57303I SUBTASK nn, DETACH COMPLETED

Explanation: This message is informational. nn = subtask number

User response: No action is required.

CKZ57307E NO TASK CONTROL BLOCK

Explanation: This is an error WTO.

User response: Call IBM Software Support.

CKZ57308E NO JOB CONTROL BLOCK

Explanation: This is an error WTO.

User response: Call IBM Software Support.

CKZ57309E NO SAVE AREA

Explanation: This is an error WTO.

User response: Call IBM Software Support.

CKZ57310E NO VIRTUAL STORAGE

Explanation: This is an error WTO.

User response: Increase the region size and resubmit the job.

CKZ60001I SUBTASK nn, INITIALIZATION COMPLETED

Explanation: This message is informational. nn = subtask number

User response: No action is required.

CKZ60002E SUBTASK nn, ESTAE INIT ERROR, RC=rrr

Explanation: This is an internal error. nn = subtask number rrr = return code

User response: Call IBM Software Support.

CKZ60003I SUBTASK nn, NUMBER OF DATASETS PER COMMAND INCREASED TO ddd

Explanation: This message is informational. It occurs because the number of data sets per DSS command is low and must be increased due to data set extensions or a clone table. nn = subtask number ddd = decimal number of the new value

User response: Consider raising the PARMLIB parameter DSNS_PER_COPY to the maximum of 255.

CKZ60004E SUBTASK nn, DEQUEING ERROR,RC=rrr

Explanation: This is an internal error. nn = subtask number rrr = return code

User response: Call IBM Software Support.

CKZ60005E ERROR TAKING SVC DUMP: rrr

Explanation: This is an error WTO. rrr = SDUMP return code

User response: Call IBM Software Support.

CKZ60006I SUBTASK nn, DATASET ddd HAS BEEN QUEUED FOR EXTENSION OVERFLOW

Explanation: This message is informational. It occurs because a primary DSN and all its extensions or clone table will not fit into the user specified data sets that may be sent to DSS during this call. nn = subtask number ddd = source data set name

User response: Consider using the default values for DSNS_PER_COPY and DSS_COPY_COMMANDS. This will reduce the occurrences of requeueing.

CKZ60007I Subtask *subtask_number*, purge request, ID=*request_type*, XEH=*pointer_to_request_buffer*

Explanation: This message is informational. It indicates which subtask requests have been purged due to an error.

User response: No action is required.

CKZ60009E ABEND 400,REASON=9

Explanation: This is an error WTO. It precedes an abend.

User response: Call IBM Software Support.

CKZ60010E ABEND 400,REASON=10

Explanation: This is an error WTO. It precedes an abend.

User response: Call IBM Software Support.

CKZ60011E ESTAE NO TASK DESCRIPTOR ELEMENT

Explanation: This is an error WTO.

User response: Call IBM Software Support.

CKZ60012E ESTAE NO CONTROL BLOCK

Explanation: This is an error WTO.

User response: Call IBM Software Support.

CKZ60013E ESTAE NO SAVE AREA

Explanation: This is an error WTO.

User response: Call IBM Software Support.

CKZ60014E ERROR TAKING SVC DUMP: rrr

Explanation: This is an error WTO. rrr = SDUMP return code

User response: Call IBM Software Support.

CKZ60015E ERROR TAKING SVC DUMP: rrr

Explanation: This is an error WTO. rrr = SDUMP return code

User response: Call IBM Software Support.

CKZ61101E SOURCE DB2 VERSION vv IS NOT SUPPORTED

Explanation: This is a user error. vv = DB2 version number

User response: Run on a supported DB2 version.

CKZ61102E SUBTASK nn, DB2 CAF INIT ERROR, UNABLE TO LOAD MODULE mmm

Explanation: This is a probable user or DB2 error. nn = subtask number mmm = name of DB2 load module

User response: Ensure SDSNLOAD is in steplib. If it is, contact your systems programmer.

CKZ61103W SUBTASK nn, DDF HAS FAILED TO CONNECT TO DB2 SUBSYSTEM tttt, TRYING TCPIP

Explanation: This is a warning that DDF was specified, but does not connect. TCPIP will be tried next. nn = subtask number tttt= target subsystem

User response: Ensure the DDF parameters are correct and that DDF is available to the target subsystem.

CKZ61104E SUBTASK nn, CANNOT CONNECT TO DB2 tttt, PARM FOR sss IS NOT AVAILABLE

Explanation: This is a probable user or DB2 error. nn = subtask number tttt= target subsystem sss = parm that is not available to TCPIP

User response: Ensure CAF, DDF or TCPIP may be used for the remote connect. If all the input parms are correct, contact your systems programmer.

CKZ61105E SUBTASK nn, DDF IN USE, BUT NO TCPIP CONNECTION AVAILABLE; DB2 COMMANDS CANNOT BE ISSUED ON THE TARGET SUBSYSTEM ssss. USER MUST MANUALLY START AND STOP ALL TARGET OBJECTS, HOWEVER PARMS INDICATE COMMANDS ARE REQUIRED.

Explanation: This is a probable user or DB2 error. nn = subtask number tttt= target subsystem

User response: Ensure TCPIP is available from the source to the target subsystem. If it is and all input parms are correct, contact your systems programmer. The workaround is to specify AUTO-STOP-TARGET-SPACE(N) and do the DB2 stop commands manually on the target subsystem.

CKZ61106E SUBTASK nn, TCPIP CONNECTION TO THE WRONG SUBSYSTEM, EXPECTING eeee, CONNECTED TO ssss

Explanation: This is a user error. The TCPIP server at the IP in the COPY command is not connected to the correct target subsystem for this source job. nn = subtask number eeee= target subsystem expecting from the TCPIP server ssss= target subsystem received from the TCPIP server

User response: Change the IP on the copy command to a TCPIP server connected to the target subsystem or change the LSSID of the current TCPIP server. Resubmit the server job if required.

CKZ61107I SUBTASK nn, REMOTE CONNECT USING CAF HAS FAILED, BUT THERE IS NO DDF LOCATION SPECIFIED IN THE COPY PARMS, TRYING TCPIP

Explanation: This is a user error if DDF is intended to be used for the remote connect. nn = subtask number

User response: Add the LOCATION subcommand to the COPY command if DDF is to be used. If a userid and password are needed for the DDF connect, add them to the COPY command also.

CKZ61108I SUBTASK NN, LOCAL DB2 SUBSYSTEM IN COMPATIBILITY MODE PER DB2

Explanation: This message is informational. nn = subtask number

User response: No action is required.

CKZ61110I SUBTASK nn, DDF IN USE, TCPIP CONNECTION MADE TO DB2 SUBSYSTEM sss

Explanation: This message is informational. DDF is used for target SQL. TCPIP is required for DB2 commands and data set processing. nn = subtask number sss = target DB2 subsystem

User response: No action is required.

CKZ61201E SUBTASK nn, DB2 SUBSYSTEM ssss
NOT ACTIVE, CAFRC=hhh

Explanation: The CAF return indicates the DB2 subsystem is not active. nn = subtask number ssss = DB2 subsystem hhh = DB2 call attach facility return code

User response: Activate the subsystem and resubmit the job if the subsystem is specified correctly. If not, then correct the source or target subsystem parameter and resubmit the job.

CKZ61202E SUBTASK nn, DB2 SUBSYSTEM ssss
NOT DEFINED, CAFRC=hhh

Explanation: The CAF return indicates the DB2 subsystem is not defined. nn = subtask number ssss = DB2 subsystem hhh = DB2 call attach facility return code

User response: Specify the source and target subsystems correctly and resubmit the job.

CKZ61203E SUBTASK nn, DB2 SUBSYSTEM ssss
CAF ERROR USING FUNCTION fff,
RC=rrr, RS=sss

Explanation: The CAF error return is unexpected. nn = subtask number ssss = DB2 subsystem fff = call attach facility function attempted rrr = call attach facility return code sss = call attach facility reason code

User response: Contact your system programmer.

CKZ61204I SUBTASK nn, CONNECT TO DB2
SUBSYSTEM ssss RELEASE rrr

Explanation: This message is informational. nn = subtask number ssss = DB2 subsystem rrr = DB2 release number

User response: No action is required.

CKZ61206I SUBTASK nn, DISCONNECT FROM
DB2 SUBSYSTEM ssss

Explanation: This message is informational. nn = subtask number ssss = DB2 subsystem

User response: No action is required.

CKZ61207E SUBTASK nn, OPEN PLAN ppp ON
DB2 SUBSYSTEM ssss HAS FAILED,
PLAN UNAVAILABLE, RC=rrr

Explanation: The CAF return indicates the DB2 open plan failed, as it is not available. nn = subtask number ppp = plan name from PARMLIB ssss = DB2 subsystem rrr = return code

User response: Bind the correct plan and resubmit the job.

CKZ61208E SUBTASK nn, SUBSYSTEM ssss CAF
REQUEST ERROR, PLAN PPP
FUNCTION fff, RC=rrr, RS=sss

Explanation: The CAF return indicates the DB2 open plan failed. as it is not available. nn = subtask number ssss = DB2 subsystem ppp = plan name fff = call attach facility function rrr = call attach facility return code sss = call attach facility reason code

User response: Bind the correct plan and resubmit the job.

CKZ61209I SUBTASK nn, PLAN ppp OPEN ON
DB2 SUBSYSTEM ssss

Explanation: This message is informational. nn = subtask number ppp = plan name ssss = DB2 subsystem

User response: No action is required.

CKZ61210I SUBTASK nn, PLAN ppp CLOSE, DB2
SUBSYSTEM ssss

Explanation: This message is informational. nn = subtask number ppp = plan name ssss = DB2 subsystem

User response: No action is required.

CKZ61211I SUBTASK nn, DB2 COMMAND
ISSUED ON ssss: ccc

Explanation: This message is informational. nn = subtask number ssss = DB2 subsystem ccc = text of the DB2 command

User response: No action is required.

CKZ61212I SUBTASK nn, DB2 COMMAND
REISSUED

Explanation: This message is informational. The previous command had to be reissued as the response area was not large enough. nn = subtask number

User response: No action is required.

CKZ61213E SUBTASK nn, DB2 COMMAND
ERROR, RC=rrr, RS=sss

Explanation: The CAF return indicates a secondary error. This error is returned in field IFCADD. nn = subtask number rrr = call attach facility return in IFCARC1 sss = call attach facility reason in IFCARC2

User response: Contact your systems programmer. See the DB2 Codes manual for the return and reason code values.

CKZ61214E SUBTASK nn, DB2 COMMAND
ERROR, NOT AUTHORIZED TO
ISSUE COMMANDS

Explanation: The CAF return indicates CKZ can not issue commands. This error is returned in the IFCADD field as X'20' nn = subtask number

User response: Contact your systems programmer.

CKZ61215E SUBTASK nn, DB2 COMMAND
ERROR, RC=rrr

Explanation: The CAF return indicates a secondary error. This error is returned in field IFCADD. nn = subtask number rrr = call attach facility return in IFCADD

User response: Contact your systems programmer. See the DB2 Codes manual for the return code value.

CKZ61221I SUBTASK nn, DB2 SUBSYSTEM ssss, tt
ddd.sss ALL PARTITIONS ALREADY
STOPPED

Explanation: This message is informational. nn = subtask number ssss = DB2 subsystem tt = space type (table space or indexspace) ddd = database name sss = space name

User response: No action is required.

CKZ61222I SUBTASK nn, DB2 SUBSYSTEM ssss
STOP SUCCESSFUL FOR tt ddd.sss
(ALL PARTITIONS)

Explanation: This message is informational. nn = subtask number ssss = DB2 subsystem tt = space type (table space or indexspace) ddd = database name sss = space name

User response: No action is required.

CKZ61223I SUBTASK nn, DB2 CMD RESP ON
ssss, DSN=dsname STAT=SSS

Explanation: This message is informational. nn = subtask number ssss = DB2 subsystem sss = status returned from DISPLAY DB2 command

User response: No action is required.

CKZ61224I SUBTASK nn, DB2 SUBSYSTEM ssss, tt
ddd.sss PARTITION ppp, ALREADY
STOPPED

Explanation: This message is informational. nn = subtask number ssss = DB2 subsystem tt = space type (table space or indexspace) ddd = database name sss = space name ppp = partition

User response: No action is required.

CKZ61225I SUBTASK nn, DB2 SUBSYSTEM ssss, tt
ddd.sss ALREADY STOPPED

Explanation: This message is informational. nn = subtask number ssss = DB2 subsystem tt = space type (table space or indexspace) ddd = database name sss = space name

User response: No action is required.

CKZ61226E SUBTASK nn, DB2 SUBSYSTEM ssss, tt
ddd.sss PARTITION ppp,
UNEXPECTED STATUS uuu, RC=rrr,
RS=aaa

Explanation: This message indicates a status other than STOP, STOPP, RO and RW was received. nn = subtask number ssss = DB2 subsystem tt = space type (table space or indexspace) ddd = database name sss = space name ppp = partition uuu = unexpected status value rrr = return code aaa = reason code

User response: Change the status of the space(s) and resubmit the job to copy every space selected for copy. To just copy the spaces with the desired status, change the parms such that MAX_COPY_RC=8 and MAX_RC=4. All spaces not copied will have a copy RC=8 in the Status Report at job end.

CKZ61227E SUBTASK nn, DB2 SUBSYSTEM ssss, tt
ddd.sss UNEXPECTED STATUS uuu,
RC=rrr, RS=aaa

Explanation: This message indicates a status other than STOP, STOPP, RO or RW was received. nn = subtask number ssss = DB2 subsystem tt = space type (table space or indexspace) ddd = database name sss = space name uuu = unexpected status value rrr = return code aaa = reason code This could happen if for example if a LOB table space is in ICOPY status.

User response: Change the status of the space(s) and resubmit the job to copy every space selected for copy. To just copy the spaces with the desired status, change the parms such that MAX_COPY_RC=8 and MAX_RC=4. All spaces not copied will have a copy RC=8 in the Status Report at job end.

CKZ61228E SUBTASK nn, DB2 COMMAND
RESPONSE, RC=rrr, RS=aaa

Explanation: This message indicates an unexpected status was received. See CKZ61229E for the response. nn = subtask number rrr = return code aaa = reason code

User response: Change the status of the space and resubmit the job.

CKZ61229E SUBTASK nn, rrr

Explanation: This message is paired with CKZ61228E and prints the response from DB2. nn = subtask number rrr = DB2 response

User response: Change the status of the space and resubmit the job.

CKZ61230I SUBTASK nn, DB2 SUBSYSTEM ssss
STOP SUCCESSFUL FOR tt ddd.sss
PARTITION ppp

Explanation: This message is informational. nn = subtask number ssss = DB2 subsystem tt = space type (table space or indexspace) ddd = database name sss = space name ppp = partition

User response: No action is required.

CKZ61231I SUBTASK nn, DB2 SUBSYSTEM ssss
STOP SUCCESSFUL FOR tt ddd.sss

Explanation: This message is informational. nn = subtask number ssss = DB2 subsystem tt = space type (table space or indexspace) ddd = database name sss = space name

User response: No action is required.

CKZ61234E SUBTASK nn, DB2 COMMAND
RESPONSE, EXPECTING eee, NOT
FOUND, RC=rrr RS=sss

Explanation: This message indicates the expected status was not received. nn = subtask number eee = response expected rrr = return code sss = reason code

User response: Change the status of the space and resubmit the job.

CKZ61235E SUBTASK nn, DB2 SUBSYSTEM ssss
STOP UNSUCCESSFUL FOR tt ddd.sss
PARTITION ppp

Explanation: This message is an error. An unexpected status was returned to the DB2 display command. This data set will not be processed further. nn = subtask number ssss = DB2 subsystem tt = space type (table space or indexspace) ddd = database name sss = space name ppp = partition

User response: Change the status of the space and resubmit the job.

CKZ61236E SUBTASK nn, DB2 SUBSYSTEM ssss
STOP UNSUCCESSFUL FOR tt ddd.sss

Explanation: This message is an error. An unexpected status was returned to the DB2 display command. This data set will not be processed further. nn = subtask number ssss = DB2 subsystem tt = space type (table

space or indexspace) ddd = database name sss = space name

User response: Change the status of the space and resubmit the job.

CKZ61237I SUBTASK nn, DB2 SUBSYSTEM ssss
START NOT REQUIRED DUE TO
INITIAL STOP STATUS FOR tt ddd.sss
PARTITION ppp

Explanation: This message is informational. nn = subtask number ssss = DB2 subsystem tt = space type (table space or indexspace) ddd = database name sss = space name ppp = partition

User response: No action is required.

CKZ61238I SUBTASK nn, DB2 SUBSYSTEM ssss
START NOT REQUIRED DUE TO
INITIAL STOP STATUS FOR tt ddd.sss

Explanation: This message is informational. nn = subtask number ssss = DB2 subsystem tt = space type (table space or indexspace) ddd = database name sss = space name

User response: No action is required.

CKZ61241I SUBTASK nn, DB2 SUBSYSTEM ssss, tt
ddd.sss ALL PARTITIONS ALREADY
STARTED OR NOT BEING STARTED

Explanation: This message is informational. Note that "Not being Started" refers to those spaces that were in STOP or STOPP status when the initial display was output. nn = subtask number ssss = DB2 subsystem tt = space type (table space or indexspace) ddd = database name sss = space name

User response: No action is required.

CKZ61242I SUBTASK nn, DB2 SUBSYSTEM ssss
START SUCCESSFUL FOR tt ddd.sss
(ALL PARTITIONS)

Explanation: This message is informational. nn = subtask number ssss = DB2 subsystem tt = space type (table space or indexspace) ddd = database name sss = space name

User response: No action is required.

CKZ61243I SUBTASK nn, DB2 SUBSYSTEM ssss
INITIALLY STOPPED, tt ddd.sss NO
START REQUIRED (ALL PARTITIONS)

Explanation: This message is informational. nn = subtask number ssss = DB2 subsystem tt = space type (table space or indexspace) ddd = database name sss = space name

User response: No action is required.

CKZ61248I SUBTASK nn, DB2 SUBSYSTEM ssss
START SUCCESSFUL FOR tt ddd.sss
PARTITION ppp

Explanation: This message is informational. nn = subtask number ssss = DB2 subsystem tt = space type (table space or indexspace) ddd = database name sss = space name ppp = partition

User response: No action is required.

CKZ61249I SUBTASK nn, DB2 SUBSYSTEM ssss
START SUCCESSFUL FOR tt ddd.sss

Explanation: This message is informational. nn = subtask number ssss = DB2 subsystem tt = space type (table space or indexspace) ddd = database name sss = space name

User response: No action is required.

CKZ61253E SUBTASK nn, DB2 SUBSYSTEM ssss
START UNSUCCESSFUL FOR tt
ddd.sss PARTITION ppp

Explanation: This message is informational. nn = subtask number ssss = DB2 subsystem tt = space type (table space or indexspace) ddd = database name sss = space name ppp = partition

User response: Contact your systems programmer.

CKZ61254E SUBTASK nn, DB2 SUBSYSTEM ssss
START UNSUCCESSFUL FOR tt
ddd.sss

Explanation: This may be a DB2 problem. nn = subtask number ssss = DB2 subsystem tt = space type (table space or indexspace) ddd = database name sss = space name

User response: Contact your systems programmer.

CKZ61255E SUBTASK nn, DB2 SUBSYSTEM ssss, tt
ddd.sss BAD PARTITION NUMBER,
EXPECTING nnn, RECEIVED ooo

Explanation: This may be a catalog consistency problem. nn = subtask number ssss = DB2 subsystem tt = space type (table space or indexspace) ddd = database name sss = space name nnn = partition number ooo = partition number

User response: Resolve the inconsistency and resubmit the job.

CKZ61261I SUBTASK nn, WAITING FOR ONE OR
MORE OBJECT STATE CHANGES,
TIME REMAINING TO WAIT = ddd
SECONDS

Explanation: This message is informational. nn = subtask number ddd = decimal number

User response: No action is required.

CKZ61262E SUBTASK nn, DB2 SUBSYSTEM ssss
ccc FAILED FOR tt ddd.sss STATUS=ttt,
EXPECTED STATUS=eee, RC=rrr,
RS=zzz

Explanation: This message indicates a status other than the desired status was received. nn = subtask number ssss = DB2 subsystem ccc = DB2 command tt = space type (table space or indexspace) ddd = database name sss = space name ttt = current status value eee = expected status value rrr = return code zzz = reason code

User response: Change the status of the space and resubmit the job.

CKZ61263I SUBTASK nn, DB2 SUBSYSTEM ssss
ccc PENDING FOR tt ddd.sss
STATUS=ttt, EXPECTED STATUS=zzz

Explanation: This message indicates a status other than the desired status was received. CKZ continues to attempt the change. nn = subtask number ssss = DB2 subsystem ccc = DB2 command tt = space type (table space or indexspace) ddd = database name sss = space name ttt = current status value zzz = expected status value

User response: No action is required.

| **CKZ61266E** Subtask *subtask_number*, DB2 command
| error, DSN*TnnnI_message*.

| **Explanation:** A DB2 command response resulted in an
| error that must be resolved before continuing. For
| instance, an authorization failure may have occurred
| that inhibits further processing by DB2 Cloning Tool
| Table Space Cloning. The specific problem can be
| identified in the message output, found in the
| CKZLOG file.

| **User response:** Locate and review the information for
| the DSN*TnnnI* message in the DB2 for z/OS Messages
| documentation. Take the specified corrective action and
| resubmit the job.

CKZ61301I TABLESPACE ddd.sss PARTITION ppp
IS FILTERED FOR DEFINE NO
ATTRIBUTE

Explanation: The object and the partition are not returned to LISTDEF processing as an existing object. ddd = database sss = space ppp = partition

User response: No action is required.

CKZ61302I TABLESPACE ddd.sss IN LIST lll
 FILTERED, ALL PARTS DELETED FOR
 DEFINE NO ATTRIBUTE

Explanation: This object is not returned to LISTDEF processing as an existing object. ddd = database sss = space lll = LISTDEF list name

User response: No action is required.

CKZ61303I INDEX ccc.nnn PARTITION ppp
 FILTERED FOR DEFINE NO
 ATTRIBUTE

Explanation: This object and this partition are not returned to LISTDEF processing as an existing object. ccc = index creator nnn = index name ppp = partition number

User response: No action is required.

CKZ61304I INDEX ccc.nnn IN LIST lll FILTERED,
 ALL PARTS DELETED FOR DEFINE
 NO ATTRIBUTE

Explanation: This object and this partition are not returned to LISTDEF processing as an existing object. ccc = index creator nnn = index name lll = LISTDEF list name

User response: No action is required.

CKZ61310I ttt Catalog populate complete, Rows=rrr,
 Hits=hhh, Elapsed time=qqq

Explanation: Catalog prefetch has populated the cache. There will be one message for each object type cached. There will be a set of messages for the source, target or both. ttt = object type rrr = decimal number of rows read from the catalog hhh = decimal number of rows matching objects needed qqg = approximate time to read the catalog table in seconds

User response: No action is required.

CKZ61311I INDEX ccc.nnn IN LIST lll FILTERED
 FOR COPY

Explanation: This message is informational. The index COPY attribute does not match the value in LISTDEF. ccc = index creator nnn = index name lll = LISTDEF list name

User response: No action is required.

CKZ61312I tt ddd.sss IN LIST lll FILTERED FOR
 CLONE

Explanation: This message is informational. The CLONE attribute does not match the value in LISTDEF. tt = space type (table space or indexspace) ddd = dbname sss = space name lll = LISTDEF list name

User response: No action is required.

CKZ61313I Subtask nn, Catalog prefetch cache
 terminated

Explanation: This message is informational. It indicates that the cache was invalidated when the client disconnected. nn = subtask number

User response: No action is required.

CKZ61314I Database ddd found in the cache

Explanation: This message is informational. It indicates that the database was found in the TCPIP server cache. ddd = database name

User response: No action is required.

CKZ61315I ccc Cache in progress for database ddd

Explanation: This message is informational. ccc = catalog, source or target ddd = database name

User response: No action is required.

CKZ61316I Cache complete, all databases found

Explanation: This message is informational. No target catalog access will be required for the given set of objects. All databases are already cached.

User response: No action is required.

CKZ61317I Cache incomplete, one or more database
 were not found

Explanation: This message is informational. A target catalog access will be required for the given set of objects. All databases are not already cached.

User response: No action is required.

CKZ61318E Caching requested for ccc object in
 database ddd, but is not in the database
 list parameter

Explanation: Caching is requested for a database that is not in the database list parm in the Copy command. ccc = Src or Trg ddd = Database name

User response: Add the database to the source and/or target database list and rerun the source job.

CKZ61319E No tablespace objects found in cache

Explanation: Caching is requested but no table space objects were found.

User response: Add the database to the source and/or target database list and rerun the source job. Add LISTDEF databases to the source database list and add the translated (if any) database names to the target database list.

CKZ61321E SUBTASK nn, FATAL SQL ERROR HAS OCCURRED

Explanation: This error terminates processing. nn = subtask number

User response: See the detailed message with the DB2 SQL error.

CKZ61322I SUBTASK nn, SQL RETRY ATTEMPT IN PROGRESS, ddd RETRY(S) REMAINING

Explanation: This message is informational. nn = subtask number

User response: No action is required.

CKZ61323I SUBTASK nn, SQL RETRY ATTEMPT SUCCESSFUL

Explanation: This message is informational. nn = subtask number

User response: No action is required.

CKZ61324E SUBTASK nn, SQL RETRY ATTEMPTS TERMINATED

Explanation: This error terminates processing. nn = subtask number

User response: See the detailed message with the DB2 SQL error.

CKZ61325W SQL ERROR IN CSECT mmm

Explanation: This is a warning message. The printed SQL error follows this message. mmm = CSECT name

User response: See the detailed message with the DB2 SQL error.

CKZ61326E SUBTASK nn, DSNTIAR ERROR PRINTING SQL ERROR ON DB2 SUBSYSTEM ssss, RC=rrr

Explanation: The SQL error cannot be printed by DB2 module DSNTIAR. nn = subtask number ssss = DB2 subsystem rrr = R15 return from DSNTIAR

User response: Contact your system programmer.

CKZ61601I SUBTASK nn, CONNECT TO LOCATION lll

Explanation: This message is informational. nn = subtask number lll = DB2 location

User response: No action is required.

CKZ61603I SUBTASK nn, DISCONNECT FROM LOCATION lll

Explanation: This message is informational. nn = subtask number lll = DB2 location

User response: No action is required.

CKZ63201W SUBTASK nn, XML NOT YET SUPPORTED, tt ssss ddd.sss DISCARDED

Explanation: This is a warning that XML is not currently supported. nn = subtask number tt = space type ssss = DB2 subsystem ddd = database name sss = space name

User response: Remove the table space with the XML column from LISTDEF or run the source job with MAX_RC=4 and copy the XML space manually.

| **CKZ66901E** Subtask *subtask_number*, ACK expected from remote DB2 subsystem, receive type *receive_message_type*

| **Explanation:** An ACK was expected from the remote TCP/IP connection for the specified receive type.

| **User response:** If unable to resolve this error, contact IBM Software Support.

| **CKZ66902E** Packet format error, *ttnn,parameter1,parameter2*

| **Explanation:** There is an error in the packet received from the remote DB2 subsystem. *ttnn* indicates the error type and a decimal qualifier. Possible *tt* values include:

- | • BL: Block number receive error
- | • CK: TCP/IP packet checksum error
- | • GM: Group accessed by minilog not found
- | • MM: Space group cannot be found
- | • RM: Number of records does not match array length
- | • UN: Unknown block type received
- | • UT: Unexpected type from server
- | • XB: No blocks found
- | • XG: Group block should first from server

| **User response:** If unable to resolve this error, contact IBM Software Support.

| **CKZ66903I** *message_explanation, number_of_records*

| **Explanation:** This message indicates that the process was successful.

| **User response:** No action is required.

**CKZ67001I CONNECTION ATTEMPT TO IP
ipaddr, PORT ppp**

Explanation: This message is informational. It is issued by the TCPIP client job. Note: ipaddr is the server IP. ppp = TCPIP bound port

User response: No action is required.

**CKZ67002I CONNECTION ATTEMPT TO IP
v6ipaddr, PORT ppp**

Explanation: This message is informational. It is issued by the TCPIP client job. Note: ipaddr is the server IP. ppp = TCPIP bound port

User response: No action is required.

**CKZ67003I TCPIP CONNECTION ESTABLISHED
ON PORT ppp USING IP ipaddr**

Explanation: This message is informational. It is issued by the TCPIP client job. Note: ipaddr is the server IP. ppp = TCPIP bound port

User response: No action is required.

**CKZ67004I TCPIP CONNECTION ESTABLISHED
ON PORT ppp USING IP v6ipaddr**

Explanation: This message is informational. It is issued by the TCPIP client job. Note: ipaddr is the server IP. ppp = TCPIP bound port

User response: No action is required.

**CKZ67005I TCPIP CONNECTION, IP ipaddr,
SOCKET sss**

Explanation: This message is informational. It is issued by the TCPIP client job. Note: ipaddr is the server IP. sss = TCPIP client socket number

User response: No action is required.

**CKZ67006I TCPIP CONNECTION, IP v6ipaddr,
SOCKET sss**

Explanation: This message is informational. It is issued by the TCPIP client job. Note: ipaddr is the server IP. sss = TCPIP client socket number

User response: No action is required.

**CKZ67039E CONNECT ERROR FOR IP ipaddr,
RETCODE=rrr, ERRNO=sss**

Explanation: This message indicates a user error or a TCPIP environment error. rrr = return code sss = ERRNO is a TCPIP reason code

User response: Correct the IP address and resubmit the job or contact the systems programmer.

**CKZ67040E CONNECT TIMED OUT FOR IP
ipaddr, RETRIES EXHAUSTED,
RETCODE=rrr, ERRNO=sss**

Explanation: This message indicates the source job is unable to connect to the TCPIP server job. ipaddr = the IP address of the server job. rrr = return code sss = ERRNO is a TCPIP reason code

User response: Verify that a TCPIP server job is waiting for a connect at the IP address and port specified in the source job. Note: the server job must be active when the source job is started. Contact the systems programmer is still unable to make the connection.

**CKZ67041E CONNECT ERROR FOR IP v6ipaddr,
RETCODE=rrr, ERRNO=sss**

Explanation: This message indicates a user error or a TCPIP environment error. v6ipaddr = the IPv6 address trying to connect to rrr = return code sss = ERRNO is a TCPIP reason code

User response: Correct the IP address and resubmit the job or contact your systems programmer.

**CKZ67042E CONNECT TIMED OUT FOR IP
v6ipaddr, RETRIES EXHAUSTED,
RETCODE=rrr, ERRNO=sss**

Explanation: This message indicates the source job is unable to connect to the TCPIP server job. v6ipaddr = the IPv6 address of the server job. rrr = return code sss = ERRNO is a TCPIP reason code

User response: Verify that a TCPIP server job is waiting for a connect at the IP address and port specified in the source job. Note: the server job must be active when the source job is started. Contact the systems programmer is still unable to make the connection.

**CKZ67111I TCPIP Client Disconnected,
Socket=socket_number**

Explanation: This message is informational. It is issued by the TCP/IP server job.

User response: No action is required.

**CKZ67121E UNABLE TO ACCEPT CONNECTION
WITH PORT ppp, RETCODE=rrr,
ERRNO=sss**

Explanation: This message indicates a probable TCPIP error. ppp = TCPIP bound port rrr = return code sss = ERRNO is a TCPIP reason code

User response: Contact your system programmer.

CKZ67124I BIND CALL SUCCESSFUL, PORT ppp

Explanation: This message is informational. It is issued by the TCPIP server job. ppp = TCPIP bound port

User response: No action is required.

CKZ67125I LISTEN CALL SUCCESSFUL, PORT PPP

Explanation: This message is informational. It is issued by the TCPIP server job. ppp = TCPIP bound port

User response: No action is required.

CKZ67126I TCPIP WAITING FOR CLIENT CONNECT ON SERVER PORT ppp

Explanation: This message is informational. It is issued by the TCPIP server job after initialization and after a client disconnect. ppp = TCPIP port waiting for connect

User response: No action is required.

CKZ67127I TCPIP CLIENT CONNECT FROM IP ipaddr

Explanation: This message is informational. It is issued by the TCPIP server job.

User response: No action is required.

CKZ67128I ACCEPT CALL SUCCESSFUL, IP ipaddr, NEW SOCKET sss

Explanation: This message is informational. It is issued by the TCPIP server job. sss = socket allocated for this connect

User response: No action is required.

CKZ67129I TCPIP CLIENT CONNECT FROM IP v6ipaddr

Explanation: This message is informational. It is issued by the TCPIP server job.

User response: No action is required.

CKZ67130I ACCEPT CALL SUCCESSFUL, IP v6ipaddr, NEW SOCKET sss

Explanation: This message is informational. It is issued by the TCPIP server job. sss = socket allocated for this connect

User response: No action is required.

CKZ67134I TCPIP GETHOSTBYNAME FOR DOMAIN NAME ddd AND HOST NAME hhh

Explanation: This message is informational. It is issued by the TCPIP client job. ddd = TCPIP domain name hhh = TCPIP host name

User response: No action is required.

CKZ67136I TCPIP SERVER CONNECTION NOT AVAILABLE, RETRY IN ONE MINUTE, ddd RETRIES REMAINING

Explanation: This message is informational. It is issued by the TCPIP client job. ddd = decimal number

User response: No action is required.

CKZ67139E UNABLE TO LISTEN ON PORT ppp, RETCODE=rrr, ERRNO=sss

Explanation: This message indicates a user error or a TCPIP environment error. ppp = TCPIP bound port rrr = return code sss = ERRNO is a TCPIP reason code

User response: Correct the port number and resubmit the job or contact your systems programmer.

CKZ67140E TCPIP server receive block error, fff, received rrr expecting xxx

Explanation: This message indicates an error occurred with the TCP/IP block number received from the client. This may be a TCP/IP transmission error or an internal programming error. fff = the function being performed rrr = block number received xxx = block number expecting

User response: If unable to resolve this error, contact IBM Software Support.

CKZ67141E TCP/IP server receive format error, fff, RC=rrr, RS=sss

Explanation: This message indicates an error occurred when the TCP/IP server validated the data received from the client. This may be a TCP/IP transmission error or an internal programming error. fff = the function being performed rrr = return code sss = reason code

User response: If unable to resolve this error, contact IBM Software Support.

CKZ67142W Database ddd removed from the cache

Explanation: This message is a warning. The database was removed from the TCP/IP server cache due to an error. This may be a TCP/IP transmission error or an internal programming error. ddd = database name

User response: If unable to resolve this error, contact IBM Software Support.

CKZ67143I TCP/IP socket shutdown, IP *ip_address*,
socket=*socket_number_used*.

Explanation: This message is informational. It is issued by the TCP/IP server job.

User response: No action is required.

CKZ67144E Shutdown error, RETCODE=TCP/
IP_return_code, ERRNO=TCP/
IP_error_number

Explanation: A TCP/IP error has occurred and the connection has been shut down.

User response: If unable to resolve this error, contact IBM Software Support.

CKZ67145E TCP/IP server received block with a
format error processing log pages,
reason_code

Explanation: An invalid TCP/IP header was received. The connection has been shut down.

User response: If unable to resolve the error, contact IBM Software Support.

CKZ67401E Subtask *subtask_number*, TCP/IP server
DDL create error.

Explanation: The TCP/IP server has issued a DDL statement that ended with an SQL error.

User response: If unable to resolve this error, contact IBM Software Support.

CKZ71000I Source subsystem *DB2_subsystem* is not a
data sharing member

Explanation: This message is informational.

User response: No action is required.

CKZ71001I Source subsystem *DB2_subsystem* is a
member of data sharing group
group_name

Explanation: This message is informational.

User response: No action is required.

CKZ71002I Data sharing
member=*data_sharing_member_name*,
ID=*data_sharing_ID*,
SSID=*DB2_subsystem*,
CMDPRFX=*command_prefix*,
STATUS=*status_of_this_member*,
DB2V=*DB2_version_of_this_member*,
SYS=*z/OS_system_name*

Explanation: This message is informational.

User response: No action is required.

CKZ71003I System checkpoint taken for subsystem
DB2_subsystem, system
z/OS_subsystem_name

Explanation: This message is informational.

User response: No action is required.

CKZ71004I Unable to acquire new checkpoint RBA,
using current value

Explanation: This message is informational.

User response: No action is required.

CKZ71005E Unable to find a checkpoint record

Explanation: The checkpoint record could not be found in the BSDS.

User response: Call IBM Software Support if unable to resolve this error.

CKZ71006E Checkpoint record format error

Explanation: The checkpoint record is in error.

User response: Call IBM Software Support if unable to resolve this error.

CKZ71007I RBA | LRSN used for log apply is
value_in_hex ddname

Explanation: This message is informational. *ddname* is for the BSDS used.

User response: No action is required.

CKZ71008I BSDS data sharing
member=*data_sharing_member_name*,
ID=*data_sharing_ID*, group=*group_name*,
BSDS01=*BSDS01_data_set_for_this_ID*,
BSDS02=*BSDS02_data_set_for_this_ID*

Explanation: This message is informational. *ddname* is for the BSDS used.

User response: No action is required.

CKZ71009E Unable to find any data sharing member
records in the BSDS

Explanation: The BSDS is in error. The BSDS data sets may be specified incorrectly.

User response: Call IBM Software Support if unable to resolve this error.

CKZ71010E No DATA-SHARING-MEMBERS command found for member ID=*data_sharing_ID*

Explanation: Unable to find all the information about this member.

User response: Supply the DATA-SHARING-MEMBERS command and rerun the job.

CKZ71011E Unable to find root record in the BSDS.

Explanation: This message indicates an error processing the BSDS data set.

User response: If unable to resolve this error, contact IBM Software Support.

CKZ71012I Using RBA/LRSN_length byte RBA/LRSN, flag=*DSNJCNVT flag_in_hex*.

Explanation: This message is informational. The RBA/LRSN length (6 or 10 bytes) used by log apply is provided in the message.

User response: No action is required.

CKZ71013E LOGAPPLY not yet supported on subsystems with 10-byte RBA/LRSN enabled.

Explanation: Log apply functionality is not available on subsystems with 10-byte RBA/LRSN enabled.

User response: Disable LOG-APPLY in the COPY command and rerun the source job.

CKZ71101I Subtask *subtask_number*, QUIESCE output open OK for DDname *ddname*

Explanation: This message is informational.

User response: No action is required.

CKZ71102I Subtask *subtask_number*, QUIESCE commands to follow ...

Explanation: This message is informational.

User response: No action is required.

CKZ71103I Subtask *subtask_number*, *message_text*

Explanation: This message is informational.

User response: No action is required.

CKZ71104E Subtask *subtask_number*, DSNUTILB attach error, RC=*return_code*, RS=*reason_code*

Explanation: Unable to start DSNUTILB.

User response: Call IBM Software Support if unable to resolve this error.

CKZ71105I Subtask *subtask_number*, table spaces quiesced at RBA/LRSN = *quiesce_point*

Explanation: This message is informational.

User response: No action is required.

CKZ71106I Subtask *subtask_number*, clone table spaces quiesced at RBA/LRSN = *quiesce_point*

Explanation: This message is informational.

User response: No action is required.

CKZ71107E Subtask *subtask_number*, DSNUTILB execute error, RC=*return_code*, RS=*reason_code*, text to follow...

Explanation: Unable to start DSNUTILB.

User response: Call IBM Software Support if unable to resolve this error.

CKZ71108I Subtask *subtask_number*, QUIESCE input open OK

Explanation: This message is informational.

User response: No action is required.

CKZ71109I Subtask *subtask_number*, QUIESCE messages to follow

Explanation: This message is informational.

User response: No action is required.

CKZ71110I Subtask *subtask_number*, text_of_QUIESCE_messages

Explanation: This message is informational.

User response: No action is required.

CKZ71400E Log apply requested data set processing; however, no data set by that name was found, DSN=*dsname*

Explanation: The data set listed in the message was not found.

User response: Contact IBM Software Support if unable to resolve this error.

CKZ71401E Log apply error return, *return_code*, *exit_name*, *dsname*

Explanation: This message is an error.

CKZ71402E • CKZ76003I

| **User response:** Contact IBM Software Support if
| unable to resolve this error.

| **CKZ71402E** PGM(SRCIMCPY) specified, copy
| started

| **Explanation:** This message indicates the start of
| cloning using image copies.

| **User response:** No action is required.

| **CKZ71403E** Subtask *subtask_number*,
| PGM(SRCIMCPY) specified, open for
| the data set already processed

| **Explanation:** This message is an error.

| **User response:** Contact IBM Software Support.

| **CKZ71404E** Subtask *subtask_number*,
| PGM(SRCIMCPY) specified, close for
| the data set already processed

| **Explanation:** This message is an error.

| **User response:** Contact IBM Software Support.

| **CKZ71501I** Subtask *subtask_number*, REBUILD
| output open OK for DD-NAME *ddname*

| **Explanation:** This message is informational.

| **User response:** No action is required.

| **CKZ71502I** Subtask *subtask_number*, REBUILD
| commands to follow...

| **Explanation:** This message is informational.

| **User response:** No action is required.

| **CKZ71503I** Subtask *subtask_number*, *message_text*

| **Explanation:** This message is informational.

| **User response:** No action is required.

| **CKZ71504E** Subtask *subtask_number*, DSNUTILB
| attach error, RC=*return_code*,
| RS=*reason_code*

| **Explanation:** This message is informational.

| **User response:** If unable to resolve this error, contact
| IBM Software Support.

| **CKZ71507E** Subtask *subtask_number*, DSNUTILB
| execute error, RC=*return_code*,
| RS=*reason_code* Text to follow...

| **Explanation:** This message is an error.

| **User response:** If unable to resolve this error, contact
| IBM Software Support.

| **CKZ71508I** Subtask *subtask_number*, REBUILD input
| open OK

| **Explanation:** This message is informational.

| **User response:** No action is required.

| **CKZ71509I** Subtask *subtask_number*, REBUILD
| messages to follow

| **Explanation:** This message is informational.

| **User response:** No action is required.

| **CKZ71510I** Subtask *subtask_number*, *message_text*

| **Explanation:** This message is informational.

| **User response:** No action is required.

| **CKZ71511I** Utility *utility_name*, RC=*return_code*

| **Explanation:** This message is informational.

| **User response:** No action is required.

| **CKZ71512W** Utility *utility_name*, Bad
| Return=*return_code*

| **Explanation:** This message is a warning. The utility
| that issued the warning is listed in the message, along
| with a return code.

| **User response:** Determine the cause from the
| preceding utility messages. Correct the problem and
| resubmit the job.

| **CKZ71513E** Utility *utility_name*, Bad
| Return=*return_code*

| **Explanation:** This message is an error. The utility that
| issued the error is listed in the message, along with a
| return code.

| **User response:** Determine the cause from the
| preceding utility messages. Correct the problem and
| resubmit the job.

| **CKZ71514E** Utility *utility_name*, *message_text*

| **Explanation:** This message is an error. The utility that
| issued the error is listed in the message, along with
| error message text.

| **User response:** Determine the cause from the utility
| messages. Correct the problem and resubmit the job.

| **CKZ76003I** One or more data sets in this log apply
| group have no log records, MLDSn is
| *name_of_minilog_data_set*

| **Explanation:** This message is informational.

| **User response:** No action is required.

CKZ76004E Subtask *subtask_number*, pending page already read and not processed, GrpPendPage=*hex_value_of_group_pending_page*, PGNUM=*hex_value_of_current_page*.

Explanation: An error has occurred during log apply processing.

User response: If unable to resolve the error, contact IBM Software Support.

CKZ76005I Subtask *subtask_number*, processing skipped per runtime repository, DSN=*data_set_name*.

Explanation: This message is informational. The data set listed in the messages processed in a previous target job.

User response: No action is required.

CKZ76006E Subtask *subtask_number*, bad eyecatcher, C call, first_8_chars_of_eyecatcher

Explanation: This is an internal error.

User response: Call IBM Software Support.

CKZ76007I Subtask *subtask_number*, data set=*data_set_name*, target data set already processed per repository.

Explanation: The runtime repository indicates that this data set has already been processed by a previous target job.

User response: No action is required.

CKZ76008E Subtask *subtask_number*, data set=*data_set_name*, runtime repository access error, request=*request_text*, response=*response_text*.

Explanation: An error occurred during VSAM processing.

User response: If unable to resolve the error, contact IBM Software Support.

CKZ76009I Subtask *subtask_number*, DSN(*data_set_name*), nodes found =*number_of_XML nodes_found_in_hex*, strings changed=*number_of_XML string IDs_changed_in_hex*

Explanation: This message provides information about XML processing.

User response: No action is required.

CKZ76021I SUBTASK nn, DSN(DSNAME), SYNCDB2 COMPLETED WITH NO ERRORS

Explanation: This message is informational. nn = subtask number

User response: No action is required.

CKZ76022I SUBTASK nn, DSN(dsname), PAGE READS=ddd, PAGE WRITES ddd

Explanation: This message is informational. nn = subtask number ddd = decimal count

User response: No action is required.

CKZ76031I THE FOLLOWING SQL STATEMENT SUBMITTED FOR EXECUTION...

Explanation: This message is informational. Message CKZ76032I has the statement.

User response: No action is required.

CKZ76032I sss

Explanation: This message is informational. sss = SQL statement

User response: No action is required.

CKZ76034E SUBTASK nn, UNABLE TO OPEN ddname DD FOR SQL EXECUTE PROCESSING, RC=rrr, RS=sss

Explanation: This message indicates an error trying to open the DD with the SQL to be executed on the target subsystem. nn = subtask number rrr = return code sss = reason code

User response: Ensure the DD specified in the source job copy command via SLOUT-DDN matches the DD specified in the target job. If so, this is probably a problem with the file itself. Correct the data set error and resubmit the job. If not, change the DD in the target job JCL to match the source job DD for SLOUT.

CKZ76035E SUBTASK nn, DSN(dsname), SYNCDB2 COMPLETED WITH ERROR(S)

Explanation: This message indicates an error trying to change the source data set IDs to the target IDs. nn = subtask number

User response: See data set processing for the specific error.

CKZ76040E Subtask *subtask_number*, page out of order, last page applied=*last_page*, next page to apply=*next_page*, current page to be updated=*current_page*, SpID=*space_ID*

Explanation: This message is a warning. *last_page* is the last page in hex to be applied. *next_page* is the next page in hex to be applied. *current_page* is the current page to be updated (PGNUM+1).

User response: Call IBM Software Support.

CKZ76092I Subtask *subtask_number*, log apply call_type call successful

Explanation: This message is informational.

User response: No action is required.

CKZ76093W Subtask *subtask_number*, log apply call_type call warning, RC=*return_code*

Explanation: This is a warning. Normally the return code is 4.

User response: Call IBM Software Support if unable to resolve this error.

CKZ76094E Subtask *subtask_number*, log apply call_type call failure, RC=*return_code*

Explanation: This is an error. Normally the return code is 8 or 12.

User response: Ensure that the minilog data set can be allocated if this is an OPEN call type. Call IBM Software Support if unable to resolve this error.

CKZ76098I Subtask *subtask_number*, CKZG-C, message_text

Explanation: A message from a page call. It could be informational, a warning or an error.

User response: No action is required, unless the call ends with an error.

CKZ76216I SYSTEM PAGE FOR ODBRECS

Explanation: This message is informational.

User response: No action is required.

CKZ76700E Subtask *subtask_number*, DDL generation catalog read error

Explanation: This is an error.

User response: Call IBM Software Support if unable to resolve this error.

CKZ76701I DDL generator completed, okay

Explanation: This message is informational.

User response: No action is required.

CKZ76702W DDL generator completed, with warning(s), RC=*return_code*, RS=*reason_code*

Explanation: This is a warning message.

User response: Call IBM Software Support if unable to resolve this error.

CKZ76703E DDL generator completed, with error(s), RC=*return_code*, RS=*reason_code*

Explanation: This is a warning message.

User response: Call IBM Software Support if unable to resolve this error.

CKZ76704I DDL write to DD *ddname*, okay

Explanation: This message is informational.

User response: No action is required.

CKZ76705E Invalid DDL generator input *input_string*

Explanation: This is an error message.

User response: Call IBM Software Support if unable to resolve this error.

CKZ76706I DDL DD *ddname*, open okay

Explanation: This message is informational.

User response: No action is required.

CKZ76707I *number_of_records* DDL records output to DD *ddname*

Explanation: This message is informational.

User response: No action is required.

CKZ76708E DDL DD *ddname*, open error, RC=*return_code*, RS=*reason_code*

Explanation: This is an error message.

User response: Call IBM Software Support if unable to resolve this error.

CKZ76709I No DDL written to DD *ddname*

Explanation: This message is informational.

User response: No action is required.

CKZ76710E No DDL written for DDL(G) or DDL(A) commands as no DDL DD was available

Explanation: This is an error message.

User response: Call IBM Software Support if unable to resolve this error.

CKZ76711E No records in DDL input DD *ddname*

Explanation: This is an error message.

User response: Call IBM Software Support if unable to resolve this error.

CKZ76712I *decimal_number_of_DDL_statements* DDL statements input from DD *ddname*

Explanation: This message is informational.

User response: No action is required.

CKZ76713E Subtask *subtask_number*, SQL error executing create object DDL, RC=*return_code*, RS=*reason_code*

Explanation: This is an error message.

User response: Call IBM Software Support if unable to resolve this error.

CKZ76714I *decimal_number_of_records* DDL statement(s) have been output to the target subsystem *db2_subsystem*

Explanation: This is message is informational.

User response: No action is required.

CKZ76715I DDL statement

Explanation: This is message is informational and displays a DDL statement.

User response: No action is required.

CKZ76716I *decimal_number_of_records* DDL statement(s) have been output to the target subsystem *db2_subsystem* for objects that already exist

Explanation: This message is informational.

User response: No action is required.

CKZ76717E Subtask *subtask_number*, terminate CKZ00992 error, RC=*return_code*, RS=*reason_code*

Explanation: This is an error message.

User response: Call IBM Software Support if unable to resolve this error.

CKZ76718E Subtask *subtask_number*, DDL statement length exceeded.

Explanation: Insufficient storage was acquired for a DDL statement.

User response: If unable to resolve the error, contact IBM Software Support.

CKZ76801E SUBTASK nn, UNABLE TO OPEN CKZCRXML DD FOR XML STRING PROCESSING

Explanation: This is a user error. nn = subtask number

User response: Correct the data set and resubmit the job.

CKZ76802E SUBTASK nn, NO RECORDS IN CKZCRXML DD FOR XML PROCESSING, RC=*rrr*

Explanation: This is a user error. nn = subtask number rrr = return code=

User response: Correct the data set and resubmit the job.

CKZ76803E SUBTASK nn, XML OBJECT TYPE *ooo*, NOT FOUND IN CKZCRXML DD

Explanation: This is a user error nn = subtask number ooo = the object type

User response: Correct the DDL in the data set and resubmit the job.

CKZ76804I SUBTASK nn, XML OBJECT TYPE *ooo* nnn FOUND IN CKZCRXML DD

Explanation: This message is informational. nn = subtask number ooo = the object type nnn = the object name

User response: No action is required.

CKZ76805E SUBTASK nn, SQL INSERT STATEMENT TO ADD STRING *ssss* TO CATALOG, ERROR, RC=*rrr*

Explanation: This is a probable DB2 error. nn = subtask number *ssss* = first 80 characters of the string rrr = return code

User response: Correct the error and resubmit the job.

CKZ76806I SUBTASK nn, SQL INSERT STATEMENTS FOR XML STRING IDS: *ddd* SUCCESSFUL

Explanation: This message is informational. nn = subtask number *ddd* = decimal number

User response: No action is required.

CKZ76807E SUBTASK nn, SQL INSERT STATEMENTS FOR XML STRING IDS: ddd SUCCESSFUL, eee UNSUCCESSFUL

Explanation: This is a summary message. nn = subtask number ddd = decimal number of successful INSERTs eee = decimal number of unsuccessful INSERTs

User response: See other error message(s) for the cause.

CKZ76808E SUBTASK nn, SQL ERROR CREATING XML STRING ID OBJECTS, RC=rrr, RS=sss

Explanation: This is an error message. The most likely cause is a DB2 error. nn = subtask number rrr = return code sss = reason code

User response: Correct the DB2 error and resubmit the job.

CKZ76809I SUBTASK nn, ALL SQL IN CKZCRXML DD COMMENTED, CKZ WILL USE EXISTING OBJECTS

Explanation: This message is informational. nn = subtask number

User response: No action is required.

CKZ76810I SUBTASK nn, TARGET STRING ID CREATED, SOURCE ID=hhh, TARGET ID=iii, ssss

Explanation: This message is informational. nn = subtask number hhh = hex number of source string id iii = hex number of target string id now in the catalog ssss = string value

User response: No action is required.

CKZ76811E SUBTASK nn, TARGET STRING ID hhh NOT FOUND IN CATALOG, RC=rrr, RS=sss

Explanation: This is a possible DB2 error. The INSERT to the XML column completed normally, but DB2 Cloning Tool Table Space Cloning is unable to read the String ID from the catalog. nn = subtask number hhh = target hex string id rrr = return code sss = reason code

User response: Determine if this is a DB2 error by querying the catalog. If not, call IBM Software Support.

CKZ78001I SUBTASK nn, DB2 SUBSYSTEM ssss, ALL ttt SPACES STOPPED

Explanation: This message is informational. nn = subtask number ssss = DB2 subsystem ttt = Target or Source

User response: No action is required.

CKZ78002I SUBTASK nn NO SOURCE SPACE STARTS PER AUTO-START-SOURCE-SPACE PARAMETER

Explanation: This message is informational. nn = subtask number

User response: No action is required.

CKZ78003E SUBTASK nn, BAD DATA MOVER VALUE

Explanation: This is an internal error. The input was validated during initialization, thus the value has been corrupted. nn = subtask number

User response: Call IBM Software Support.

CKZ78004I SUBTASK nn, DB2 SUBSYSTEM ssss, ALL ttt REQUIRED SPACE STARTS COMPLETED

Explanation: This message is informational. For the target subsystem, this is all the spaces. For the source subsystem, it is all the spaces that were initially started in RW or RO. nn = subtask number ssss = DB2 subsystem ttt = Target or Source

User response: No action is required.

CKZ78005E SUBTASK nn, ADRDSSU HAS ENDED WITH A RETURN CODE > 8

Explanation: This message is an error. DSS is unable to process any more copy requests. nn = subtask number

User response: Determine the problem using DSS messages and resubmit.

CKZ78006I NO STARTS DONE FOR THE TARGET OBJECTS, TO BE DONE AT THE END OF TARGET JOB

Explanation: This message is informational. The target objects remain stopped until the target job has run.

User response: No action is required.

CKZ78007E TABLE ccc.nnn AND COLUMN III, DEFAULT COLUMN=c, UNABLE TO REREAD IDENTITY COLUMN

Explanation: This is a DB2 catalog inconsistency. The column indicated it was an identity column, but it was not found in the identity column catalog table. ccc = table creator nnn = table name III = column name c = column default from SYSCOLUMNS

User response: Resolve the inconsistency and resubmit the job.

CKZ78010I SUBTASK nn, DB2 SUBSYSTEM ssss, ATTEMPTING RECOVERY OF SOURCE OBJECTS TO PRE EXECUTE STATUS

Explanation: This message is informational. nn = subtask number ssss = DB2 subsystem

User response: No action is required.

CKZ78011I SUBTASK NN, DB2 SUBSYSTEM ssss, tt ddd.sss STATUS CHANGE NOT ATTEMPTED, NO RECOVERY REQUIRED

Explanation: This message is informational. nn = subtask number ssss = DB2 subsystem tt = space type (table space or indexspace) ddd = database name sss = space name

User response: No action is required.

CKZ78012I SUBTASK nn, DB2 SUBSYSTEM ssss, tt ddd.sss ppp STATUS CHANGE NOT ATTEMPTED, NO RECOVERY REQUIRED

Explanation: This message is informational. nn = subtask number ssss = DB2 subsystem tt = space type (table space or indexspace) ddd = database name sss = space name ppp = partition

User response: No action is required.

CKZ78014I SUBTASK nn, DB2 SUBSYSTEM ssss, NO RECOVERY OF TARGET OBJECTS TO PRE EXECUTE STATUS DUE TO SIM(N) PARM SPECIFICATION

Explanation: This message is informational. nn = subtask number ssss = DB2 subsystem

User response: No action is required.

CKZ78015W SUBTASK nn, UNABLE TO CONNECT TO SSID ssss USING CAF, CAN NOT ISSUE TARGET RECOVERY COMMANDS

Explanation: This message is a warning. nn = subtask number ssss = DB2 subsystem

User response: Contact your system programmer.

CKZ78016I SUBTASK nn, DB2 SUBSYSTEM ssss, ATTEMPTING RECOVERY OF TARGET OBJECTS TO PRE EXECUTE STATUS DUE TO SIM(A) PARM SPECIFICATION

Explanation: This message is informational. nn = subtask number ssss = DB2 subsystem

User response: No action is required.

CKZ78022I SUBTASK nn, lll, ALLOCATION OK, sss

Explanation: This message is informational. nn = subtask number lll = load module name of copy utility sss = DD allocation statement

User response: No action is required.

CKZ78024I SUBTASK nn, DDNAME(ddname), ALLOCATION OK

Explanation: This message is informational. nn = subtask number

User response: No action is required.

CKZ78027I SUBTASK nn, lll, FREE OK, sss

Explanation: This message is informational. nn = subtask number lll = load module name of copy utility sss = DD allocation statement

User response: No action is required.

CKZ78030E SUBTASK nn, lll COMMAND RESPONSE PARSING ERROR RC=rrr, RS=sss

Explanation: This is an error caused by getting an unexpected value in the message string returned by the copy program. nn = subtask number lll = copy program rrr = return code sss = reason code

User response: Call IBM Software Support.

CKZ78031E SUBTASK nn, MAX_COPY_RC EXCEEDED WITH RC=rrr, RS=sss

Explanation: This error will terminate the source job as requested by MAX_COPY_RC. The DB2 Cloning Tool Table Space Cloning return code is an 8 as MAX_RC = 0. Note that copies already sent to the copy program will continue. No new commands will be issued, however. nn = subtask number rrr = return code sss = reason code

User response: Call IBM Software Support.

CKZ78032W SUBTASK nn, ERROR PARSING ttt; ONE OR MORE DSS DATA SET RETURN CODES MAY COME FROM DSS FINAL RETURN CODE

Explanation: While attempting to parse the DSS output, an error has occurred. This may prevent obtaining the return code for a particular data set. If so, the final DSS return code will be assigned to all those data sets without an explicit return code. nn = subtask number ttt = return value trying to parse

User response: Check the individual return codes as printed out in CKZPRINT.

CKZ78033W SUBTASK nn, MAX_COPY_RC EXCEEDED WITH RC=rrr, RS=sss

Explanation: This error will terminate the source job as requested by MAX_COPY_RC. The DB2 Cloning Tool Table Space Cloning return code is a 4 as MAX_RC > 0. Note that copies already sent to the copy program will continue. No new commands will be issued, however. nn = subtask number rrr = return code sss = reason code

User response: Find the DSS error message(s) in CKZPRINT output. If the messages are related to storage or other resource shortages, try reducing the number of DSS commands specified in the DSS_COPY_COMMANDS setting in the PARMLIB. The default is 24. Rerun the source job using a smaller value, such as 4. If the DSS error messages are not related to storage, resolve the DSS error or errors and resubmit the job to copy the failed data set(s).

CKZ78034W SUBTASK nn, RETURN OF 8 ENCOUNTERED FOR A DATA SET COPY

Explanation: This warning indicates a copy failed and MAX_COPY_RC = 8. Copies continue. nn = subtask number

User response: Resolve the DSS error or errors and resubmit the job to copy the failed data set(s).

CKZ78041I SUBTASK nn, DSS OUTPUT OPEN OK FOR DDNAME DDNAME

Explanation: This message is informational. nn = subtask number

User response: No action is required.

CKZ78042I SUBTASK nn, DSS COMMANDS TO FOLLOW ...

Explanation: This message is informational. See CKZ78043I for the command text. nn = subtask number

User response: No action is required.

CKZ78043I SUBTASK nn, ccc

Explanation: This message is informational. nn = subtask number ccc = command line sent to DSS

User response: No action is required.

CKZ78044I SUBTASK nn, DSS INPUT OPEN OK

Explanation: This message is informational. nn = subtask number

User response: No action is required.

CKZ78045I SUBTASK nn, DSS MESSAGES TO FOLLOW

Explanation: This message is informational. See CKZ78046I for the message text. nn = subtask number

User response: No action is required.

CKZ78046I SUBTASK nn, mmm

Explanation: This message is informational. nn = subtask number mmm = DSS output message text

User response: No action is required.

CKZ78047I SUBTASK nn, lll, COPY FROM ddd TO eee, COMPLETED, RETURN CODE IS ZERO

Explanation: This message is informational. nn = subtask number lll = copy program name ddd = source data set name eee = target data set name

User response: No action is required.

CKZ78048W SUBTASK nn, lll, COPY FROM ddd TO eee, COMPLETED WITH WARNINGS, RETURN CODE IS 4

Explanation: This message is a warning from DSS. If MAX_COPY_RC is 4 or lower, the job will terminate. nn = subtask number lll = copy program name ddd = source data set name eee = target data set name

User response: Determine the cause of the error using DSS output messages and resubmit the job.

CKZ78050E ADRDSSU has ended with a fatal error.

Explanation: This message indicates an error has been found by ADRDSSU that is severe enough to stop copy processing.

User response: If unable to determine the cause of the ADRDSSU error, contact IBM Software Support.

CKZ78051E SUBTASK nn, lll, DDNAME(ddname), RC=rrr, RS=sss, OPEN FAILED

Explanation: This message is an error with the DSS interface. nn = subtask number lll = copy program name rrr = return code sss = reason code

User response: Call IBM Software Support.

CKZ78051E SUBTASK nn, lll, DDNAME(ddname), RC=rrr, RS=sss, OPEN FAILED

Explanation: This message is an error with the DSS interface. nn = subtask number lll = copy program name rrr = return code sss = reason code

User response: Call IBM Software Support.

CKZ78052E SUBTASK nn, lll, DDNAME(ddname), RC=rrr, RS=sss, PUT FAILED

Explanation: This message is an error with the DSS interface. nn = subtask number lll = copy program name rrr = return code sss = reason code

User response: Call IBM Software Support.

CKZ78053E SUBTASK nn, lll, DDNAME(ddname), RC=rrr, RS=sss, LOAD FAILED

Explanation: This message is an error with the DSS interface. The load module could not be found. nn = subtask number lll = copy program name rrr = return code sss = reason code

User response: Contact your systems programmer.

CKZ78061W SUBTASK nn, UNABLE TO CONNECT TO SSID ssss USING CAF, CAN NOT ISSUE STOP COMMANDS

Explanation: This is a warning. CKZ will attempt copies if MAX_RC=4. nn = subtask number ssss = DB2 subsystem

User response: Contact your system programmer.

CKZ78071W SUBTASK nn, UNABLE TO CONNECT TO SSID ssss USING CAF, CAN NOT ISSUE START COMMANDS

Explanation: This is a warning. nn = subtask number ssss = DB2 subsystem

User response: Contact your system programmer.

CKZ78102I Maximum number of data sets per DSS call, max_number_in_decimal, number of data sets requested, number_of_data_sets_to_copy_in_decimal, number of DSS job steps required, number_of_DSS_steps_required.

Explanation: This message is informational. It indicates that multiple steps must be coded in the input job template.

User response: Add the number of steps to the input template(s).

CKZ78116W A JOB TEMPLATE WAS SPECIFIED, HOWEVER NO DATASETS MEET THE COPY CRITERIA

Explanation: This message is a warning. No job template was completed as no target data sets can be found.

User response: Drop the JOB-TEMPLATE command or re-specify the objects to be processed so that one or more target data sets can be copied.

CKZ78118E Input template DD ddname, object variable object_variable_name is incorrect.

Explanation: An error has occurred in the input job template.

User response: If unable to resolve this error, contact IBM Software Support.

CKZ78119I Number of source | target objects available for job template processing: TS=decimal_number_of_table-space_objects, IS=decimal_number_of_index_objects.

Explanation: This message is informational. It indicates which subtask requests have been purged due to an error.

User response: No action is required.

CKZ78120E Input template DD ddname, both data set and object variables are specified.

Explanation: An error occurred during job template input processing.

User response: If unable to resolve the error, contact IBM Software Support.

CKZ78121E Input template DD ddname, object variable variable_name was specified but no objects of that type were found

Explanation: An error occurred during job template input processing.

User response: If unable to resolve the error, contact IBM Software Support.

CKZ78122I Index Rebuild Pending for DSN=dsname

Explanation: This message is informational. A table associated with this index has been modified by data masking or log apply.

User response: Rebuild this index after target job completion.

CKZ78123E Input Template DD *ddname*, Object Variable *variable* is Invalid in the Target Job

Explanation: This message indicates an error. *ddname* is the DD containing the invalid variable and *variable* is the invalid variable in the job template.

User response: Correct the job template input variable and resubmit the job.

CKZ78124W Subtask *subtask_number*, input template DDNAME references *job_type* *missing_object* but none were found.

Explanation: A template that is referenced by the job has no objects in the job.

User response: Correct the input template and resubmit the job.

CKZ78201E SUBTASK *nn*, *mmm*, RC=*rrr*, RS=*sss*, LOAD FAIL

Explanation: This message indicates an error. Module *mmm* was not loaded. *nn* = subtask number *mmm* = module that could not be loaded *rrr* = return code *sss* = reason code

User response: This is an EMC module. Ensure the EMC modules are in the linklist or have been added to the source job steplib and resubmit the job.

CKZ78202I EMCAPI VERSION=*vvv*, RELEASE=*rrr*, LEVEL=*lll*, PTF=*ppp*

Explanation: This message is informational. *vvv* = version number *rrr* = release number *lll* = level number *ppp* = PTF number

User response: No action is required.

CKZ78203E SUBTASK *subtask_number*, *error_description*, RC=*return_code*, RS=*reason_code*

Explanation: This message indicates an error. A brief description of the error is contained in the message, as well as the return and reason codes. Because the EMCAPI call return code exceeded the value of MAX-COPY-RC, this message is considered an error.

User response: Check all EMC warning and error messages to determine the correct action to take.

CKZ78204I SUBTASK *nn*, *ttt*

Explanation: This message is informational. *nn* = subtask number *ttt* = text of the EMC message

User response: No action is required.

CKZ78205I SUBTASK *nn*, EMC MESSAGES TO FOLLOW

Explanation: This message is informational. *nn* = subtask number

User response: No action is required.

CKZ78206E SUBTASK *nn*, UNEXPECTED RETURN FROM EMC COPY, RC=*rrr*

Explanation: This message indicates an error. *nn* = subtask number *rrr* = returned code

User response: Check all EMC messages. If unable to determine from those messages what the problem is, call IBM Software Support.

CKZ78207E SUBTASK *nn*, EMCAPI HAS ENDED WITH A RETURN CODE > 8

Explanation: This message indicates an error. *nn* = subtask number

User response: Check all EMC messages. If unable to determine from those messages what the problem is, call IBM Software Support.

CKZ78208W SUBTASK *nn*, EMCAPI MESSAGE SYNC ERROR, RC=*rrr*, RS=*sss*

Explanation: This message is a warning. *nn* = subtask number *rrr* = return code *sss* = reason code

User response: Call IBM Software Support.

CKZ78209W SUBTASK *nn*, EMCAPI MESSAGE, SUMMARY RC *rrr*, Does Not Match Highest Dataset RC *ddd*

Explanation: This message is a warning. *nn* = subtask number *rrr* = return code from ESNP440I *sss* = highest return code from ESNP471I messages

User response: Call IBM Software Support.

CKZ78210W SUBTASK *nn*, ERROR PARSING *sss*; ONE OR MORE EMC DATASET RETURN CODES MAY COME FROM EMC FINAL RETURN CODE

Explanation: This message is a warning. Any copy return codes not processed correctly, will be set using ESNP440I. See CKZ78212W for the names of the data sets affected. *nn* = subtask number *sss* = string not found

User response: Call IBM Software Support.

CKZ78211E SUBTASK *nn*, ERROR PARSING *sss*,
CANNOT FIND THE FINAL RETURN
CODE

Explanation: This message indicates an error. *nn* =
subtask number *sss* = string not found

User response: Call IBM Software Support.

CKZ78212W SUBTASK *nn*, DATASET *dsname*,
CANNOT FIND RC, RC COMES FROM
THE FINAL RETURN MESSAGE,
ESNP440I

Explanation: This message is a warning. The final
return code for this data set will be set using ESNP440I.
nn = subtask number *sss* = string not found

User response: Call IBM Software Support.

CKZ78213W SUBTASK *subtask_number*,
warning_description, **RC=return_code**,
RS=reason_code

Explanation: This message is a warning. A brief
description of the warning or failure is contained in the
message, as well as the return and reason codes.
Because the EMC API call return code did not exceed
the value of MAX-COPY-RC, this message is considered
a warning.

User response: Check all EMC warning and error
messages to determine the correct action to take.

CKZ90101E RENAME MASK IS INVALID: **new
name mask**

Explanation: New name mask is invalid.

User response: Correct the error in the
RENAME-MASKS keyword.

CKZ90102E NEW NAME GREATER THAN 44
BYTES: new datasetname

Explanation: New name masks resulted in new data
set name that was greater than 44 characters.

User response: Correct the error in the
RENAME-MASKS keyword.

CKZ90103E NEW NAME IS INVALID: **new
datasetname**

Explanation: New name masks resulted in new data
set name that was invalid.

User response: Correct the error in the
RENAME-MASKS keyword.

CKZ90201W SMS CLASSES NOT RETURNED BY
ACS ROUTINES: **RC=yyyyyyyy**
REASON=zzzzzzzz **DSN=datasetname**

Explanation: Attempt to derive the new SMS classes
for a data set failed.

User response: Check job log and CKZPRINT for any
additional messages. If unable to determine the reason
for failure, contact IBM Software Support. Have
available the listing that contains this message.

CKZ90207W ERROR CALLING CKZ01HEX;
FUNCTION: function R15=nnnn

Explanation: An error occurred using CKZ01HEX to
print a record. Processing continues.

User response: Please report this message to IBM
Software Support.

CKZ90410E UNEXPECTED RETURN CODE FROM
RACROUTE

Explanation: A RACROUTE macro call got an
unexpected return code. Processing terminates.

User response: Contact IBM Software Support. Have
available the listing containing this message.

CKZ98201E Subtask *subtask_number*, **RS=reason_code**,
Invalid DIB

Explanation: This is an error message.

User response: Call IBM Software Support.

CKZ98202E Subtask *subtask_number*, **RS=reason_code**,
module_name **LOAD Failed**

Explanation: This is an error message.

User response: Call IBM Software Support if unable to
resolve this error.

CKZ98203E Subtask *subtask_number*, **RC=return_code**,
RS=reason_code, **TOKEN=LE_token**,
LERC=LE_return_code,
LERS=LE_reason_code,
LEFDBK=LE_feedback_code, **LE interface
error, call type=LE_call_type**

Explanation: This is an error message.

User response: Call IBM Software Support if unable to
resolve this error.

CKZ98204E Subtask *subtask_number*, **RC=return_code**,
RS=reason_code **TOKEN=LE_token**, **Error
Return from the DDL Generation
Processor**

Explanation: This is an error message.

User response: Call IBM Software Support if unable to resolve this error.

CKZ99000I SUBTASK *subtask_number*, DATA MASKING ENGINE HAS INITIALIZED SUCCESSFULLY

Explanation: This message is informational.

User response: No action is required.

CKZ99001E SUBTASK *subtask_number*, UNABLE TO INITIALIZE DATA MASKING ENGINE

Explanation: An error occurred during initialization of data masking engine.

User response: See previous error messages for more information.

CKZ99002E SUBTASK *subtask_name*, UNABLE TO FIND FIELD *field_name* IN THE TABLE *creator_name.table_name*

Explanation: The field doesn't exist in the table. This error happens during RI or masks initialization.

User response: Contact IBM Software Support if this error happens during RI initialization. Change field name in the mask rule definition if this error happens during masks initialization.

CKZ99003E SUBTASK *subtask_number*, UNABLE TO ADD LINK BETWEEN *parent_creator_name.parent_table_name (parent_field_name)* AND *child_creator_name.child_table_name (child_field_name)*

Explanation: An error occurred during internal adding of link between tables.

User response: See previous error messages for more information.

CKZ99004E SUBTASK *subtask_number*, MASK RULE *mask_rule-name* IS NOT ALLOWED FOR FIELD *creator_name.table_name(field_name)* WHICH HAS TYPE *field_type*

Explanation: A mask rule with designated name is not applicable for the type of designated field.

User response: Change mask rule or field name in the mask rule definition.

CKZ99005E SUBTASK *subtask_number*, AN ERROR OCCURRED WHILE ADDING *mask_rule_name* FOR *creator_name.table_name(field_name)*

Explanation: An error occurred during internal adding of mask rule.

User response: See previous error message for more information.

CKZ99006E SUBTASK *subtask_number*, *parameter_name* VALUE IS OUT OF RANGE [*minimum_value*, *maximum_value*]; RULE='mask_rule_name', FIELD='creator_name.table_name (*field_name*)'

Explanation: Parameter value is out of range.

User response: Change parameter value in the mask rule definition.

CKZ99007E SUBTASK *subtask_number*, MINIMAL (*minimum_parameter*) VALUE EXCEEDED OR EQUALED TO MAXIMAL (*maximum_parameter*) VALUE

Explanation: A MINIMAL value exceeded or equaled the MAXIMAL value.

User response: Change MINIMAL or MAXIMAL value of range in the mask rule definition.

CKZ99008E SUBTASK *subtask_number*, DESIGNATED RANGE [*minimum_parameter*, *maximum_parameter*] IN MASK RULE EXCEEDED FIELD LENGTH (*field_length*)

Explanation: The designated range in the MASK RULE exceeded field length.

User response: Change range in the mask rule definition.

CKZ99009E SUBTASK *subtask_number*, UNABLE TO LOAD USER EXIT *user_exit_name*

Explanation: Unable to load user exit.

User response: Change the user exit name in the mask rule definition.

CKZ99010E SUBTASK *subtask_number*, UNABLE TO PARSE SUPPLIED PATTERN '*pattern*'

Explanation: Unable to parse supplied mask pattern.

User response: See previous error messages for more information.

CKZ99011I SUBTASK *subtask_number*, ADDING LINK BETWEEN *parent_creator_name.parent_table_name (parent_field_name)* AND *child_creator_name.child_table_name (child_field_name)* COMPLETED SUCCESSFULLY

Explanation: This message is informational.

User response: No action is required.

CKZ99012I SUBTASK *subtask_number*, ADDING
mask_rule FOR
creator_name.table_name(field_name)
COMPLETED SUCCESSFULLY

Explanation: This message is informational.

User response: No action is required.

CKZ99013E SUBTASK *subtask_number*, UNABLE TO
FIND TABLE *creator_name.table_name*

Explanation: This is an internal error.

User response: Contact IBM Software Support.

CKZ99014E SUBTASK *subtask_number*, UNABLE TO
ADD CREATOR *creator_name*

Explanation: This is an internal error.

User response: Contact IBM Software Support.

CKZ99015E SUBTASK *subtask_number*, FIELD
field_name ALREADY EXISTS IN
creator_name.table_name

Explanation: This is an internal error.

User response: Contact IBM Software Support.

CKZ99016E SUBTASK *subtask_number*, TABLE
creator_name.table_name ALREADY
EXISTS

Explanation: This is an internal error.

User response: Contact IBM Software Support.

CKZ99017E SUBTASK *subtask_number*, EXPECTED
SYMBOL *symbol* NOT RECEIVED

Explanation: The expected symbol was not received in
a mask pattern.

User response: See error messages following this error
for more information.

CKZ99018E SUBTASK *subtask_number*, INVALID
MASK PATTERN '*mask_pattern*'

Explanation: An invalid mask pattern was
encountered.

User response: See error messages following this error
for more information.

CKZ99019E SUBTASK *subtask_number*,
UNEXPECTED STRING '*string*' HAS
BEEN RECEIVED

Explanation: An unexpected string was encountered
in a mask pattern.

User response: See error messages following this error
for more information.

CKZ99020E SUBTASK *subtask_number*, THE
SPECIFIED STRING '*error_string*' DOES
NOT CORRESPOND TO *mask_rule_name*
PATTERN

Explanation: The specified string not corresponded to
a mask pattern.

User response: Correct the string in the mask rule
definition.

CKZ99021E SUBTASK *subtask_number*, UNABLE TO
FIND CREATOR *creator_name*

Explanation: This is an internal error.

User response: Contact IBM Software Support.

CKZ99022E SUBTASK *subtask_number*, INVALID
MASK RULE NAME: '*mask_rule_name*'

Explanation: An invalid mask rule name was entered.

User response: Correct the mask rule name in the
mask rule definition.

CKZ99023E SUBTASK *subtask_number*, UNABLE TO
GET CURRENT SYSTEM DATE

Explanation: This is an internal error.

User response: Contact IBM Software Support.

CKZ99024E SUBTASK *subtask_number*, UNABLE TO
GET CURRENT SYSTEM TIME

Explanation: This is an internal error.

User response: Contact IBM Software Support.

CKZ99025E SUBTASK *subtask_number*, UNABLE TO
GET CURRENT SYSTEM TIMESTAMP

Explanation: This is an internal error.

User response: Contact IBM Software Support.

CKZ99026E SUBTASK *subtask_number*,
DESIGNATED KEY
parent_creator_name.parent_table_name
(*parent_field_name*) <=>
child_creator_name.child_table_name
(*child_field_name*) ALREADY EXISTS

CKZ99027E • CKZ99039W

Explanation: This is an internal error.

User response: Contact IBM Software Support.

CKZ99027E SUBTASK *subtask_number*, UNABLE TO ADD RULE '*mask_rule*' BECAUSE FIELD *creator_name.table_name(field_name)* INVOLVED IN RELATION

Explanation: Unable to add the rule because the field listed is involved in relation.

User response: Delete mask rule definition or change the field name in the mask rule definition.

CKZ99028E SUBTASK *subtask_number*, UNABLE TO GET CURRENT SYSTEM USER

Explanation: This is an internal error.

User response: Contact IBM Software Support.

CKZ99029E SUBTASK *subtask_number*, UNABLE TO ADD TABLE *creator_name.table_name*

Explanation: This is an internal error.

User response: Contact IBM Software Support.

CKZ99030E SUBTASK *subtask_number*, UNABLE TO ADD FIELD *creator_name.table_name(field_name)*

Explanation: This is an internal error.

User response: Contact IBM Software Support.

CKZ99031E SUBTASK *subtask_number*, INVALID FIELD TYPE: '*field_type*'

Explanation: This is an internal error.

User response: Contact IBM Software Support.

CKZ99032E SUBTASK *subtask_number*, INVALID MIE_RECfmt = *value_of_MIE_RECfmt*. 9 IS EXPECTED (_RFPAGE - PROCESS PAGE)

Explanation: This is an internal error.

User response: Contact IBM Software Support.

CKZ99033E SUBTASK *subtask_number*, POINTER TO THE MIE STRUCT IS NULL

Explanation: This is an internal error.

User response: Contact IBM Software Support.

CKZ99034E SUBTASK *subtask_number*, POINTER TO THE DATA_PAGE_HEADER STRUCT IS NULL

Explanation: This is an internal error.

User response: Contact IBM Software Support.

CKZ99035E SUBTASK *subtask_number*, CURRENT PAGE SIZE IS ZERO

Explanation: This is an internal error.

User response: Contact IBM Software Support.

CKZ99036E SUBTASK *subtask_number*, PDSN_FORMAT IS NOT DEFINED

Explanation: This is an internal error.

User response: Contact IBM Software Support.

CKZ99037E SUBTASK *subtask_number*, INVALID PDSN_FORMAT = '*pdsn_format_symbol*'. ' OR 'R' IS EXPECTED

Explanation: This is an internal error.

User response: Contact IBM Software Support.

CKZ99038W SUBTASK *subtask_number*, PGFLAGS = *hex_value_of_PGFLAGS*. CURRENT DATA PAGE WAS NOT PROCESSED

Explanation: The current data page was not processed because of a problem with PGFLAGS. Therefore, not all data was processed in the data base. PGFLAGS is internal hex value.

User response: If this error reoccurs and you would like this data page to be processed, contact IBM Software Support.

CKZ99039W SUBTASK *subtask_number*, BPAGELASTBYTEFLAG = *hex_value_of_bPageLastByteFlag*; BTPAGELASTBYTE = *hex_value_of_btPageLastByte*. CURRENT DATA PAGE WAS NOT PROCESSED

Explanation: The current data page was not processed due to problem with the pairing of bPageLastByteFlag and btPageLastByte. It must be: 1) bPageLastByteFlag = zero, btPageLastByte = 0xC5; or 2) bPageLastByteFlag = not zero, btPageLastByte = 0xD5. Other combinations are invalid. If that is the case, then not all data was processed in the data base.

User response: If masking only a portion of the data is unacceptable, contact IBM Software Support.

CKZ99040W SUBTASK *subtask_number*, PGSFLAGS = *hex_value_of_PGSFLAGS*. CURRENT RECORD WAS NOT PROCESSED

Explanation: The current record was not processed due to problem with PGSFLAGS. It means not all data was processed in the data base. PGSFLAGS is internal hex value.

User response: If masking only a portion of the data is unacceptable, contact IBM Software Support.

CKZ99041E SUBTASK *subtask_number*, TABLE NOT FOUND BY OBID = *obid*

Explanation: This is an internal error.

User response: Contact IBM Software Support.

CKZ99042I SUBTASK *subtask_number*, CURRENT "PAGE" CALL PROCESSED SUCCESSFULLY

Explanation: This message is informational.

User response: No action is required.

CKZ99043W SUBTASK *subtask_number*, CURRENT "PAGE" CALL COMPLETED WITH WARNINGS

Explanation: Warnings have been issued during processing of "PAGE" call.

User response: See previous warning messages.

CKZ99044E SUBTASK *subtask_number*, UNABLE TO PROCESS THE CURRENT "PAGE" CALL DUE TO ERRORS

Explanation: An error occurred during processing of "PAGE" call.

User response: See previous error messages.

CKZ99045I SUBTASK *subtask_number*, CURRENT "SDSN" CALL PROCESSED SUCCESSFULLY

Explanation: This message is informational.

User response: No action is required.

CKZ99046W SUBTASK *subtask_number*, CURRENT "SDSN" CALL COMPLETED WITH WARNINGS

Explanation: Warnings have been issued during processing of "SDSN" call.

User response: See previous warning messages.

CKZ99047E SUBTASK *subtask_number*, UNABLE TO PROCESS THE CURRENT "SDSN" CALL DUE TO ERRORS

Explanation: An error occurred during processing of "SDSN" call.

User response: See previous error messages.

CKZ99048E SUBTASK *subtask_number*, INVALID MIE_RECfmt = *value_of_MIE_RECfmt*. 8 IS EXPECTED (_RFPDSN - PROCESS DATA SET)

Explanation: This is an internal error.

User response: Contact IBM Software Support.

CKZ99049I SUBTASK *subtask_number*, CURRENT DATA SET IS *data_set_name*

Explanation: This message is informational.

User response: No action is required.

CKZ99050W SUBTASK *subtask_number*, INITIALIZATION COMPLETED WITH WARNINGS

Explanation: Warnings occurred during initialization of data masking engine.

User response: See previous warning messages.

CKZ99051E SUBTASK *subtask_number*, FTR_STATIC() FAULT

Explanation: This is an internal error.

User response: Contact IBM Software Support.

CKZ99052E SUBTASK *subtask_number*, FTR_CURRENT_USER() FAULT

Explanation: This is an internal error.

User response: Contact IBM Software Support.

CKZ99053E SUBTASK *subtask_number*, FTR_USER_EXIT() FAULT

Explanation: This is an internal error.

User response: Contact IBM Software Support.

CKZ99054E SUBTASK *subtask_number*, FTR_SCRAMBLE() FAULT

Explanation: This is an internal error.

User response: Contact IBM Software Support.

CKZ99055E SUBTASK *subtask_number*, MASK() FAULT

Explanation: This is an internal error.

User response: Contact IBM Software Support.

CKZ99056E SUBTASK *subtask_number*, PATTERN() FAULT

Explanation: This is an internal error.

User response: Contact IBM Software Support.

CKZ99057E SUBTASK *subtask_number*, INVALID MIE_COL_NULLS = *'value_of_MIE_COL_NULLS'*. IT MUST BE A 'Y' OR 'N'

Explanation: This is an internal error.

User response: Contact IBM Software Support.

CKZ99058E SUBTASK *subtask_number*, INVALID MASK TYPE *'mask_rule_type'* FOR FIELD *creator_name.table_name(field_name)*

Explanation: Mask with designated type is not applicable for the designated field.

User response: Correct the mask rule or field name in the mask rule definition.

CKZ99059E SUBTASK *subtask_number*, INVALID MASK PATTERN *'mask_pattern_string'* FOR FIELD *creator_name.table_name(field_name)*

Explanation: Designated mask pattern is not applicable for designated field.

User response: Correct the mask rule or field name in the mask rule definition.

CKZ99060E SUBTASK *subtask_number*, RI INITIALIZATION ERROR HAS OCCURRED

Explanation: Initialization has completed with RI initialization error.

User response: See previous error message for more information.

CKZ99061E SUBTASK *subtask_name*, INVALID MIE_RECfmt = *value_of_MIE_RECfmt*. 3 IS EXPECTED (_RFIDSN - INITIAL DATA SET)

Explanation: This is an internal error.

User response: Contact IBM Software Support.

CKZ99062E SUBTASK *subtask_name*, INVALID MIE_RECfmt = *value_of_MIE_RECfmt*. 4 IS EXPECTED (_RFTS - TABLE SPACE). RFIDSN.MIE_IDSNDNAME = *"data_set_name"*

Explanation: This is an internal error.

User response: Contact IBM Software Support.

CKZ99063E SUBTASK *subtask_name*, INVALID MIE_RECfmt = *value_of_MIE_RECfmt*. 5 IS EXPECTED (_RFTB - TABLE RECORD). RFIDSN.MIE_IDSNDNAME = *"data_set_name"*

Explanation: This is an internal error.

User response: Contact IBM Software Support.

CKZ99064W SUBTASK *subtask_name*, LENGTH OF VALUE OF PATTERN *'pattern'* IS GREATER THAN LENGTH OF COLUMN *creator_name.table_name(field_name)*

Explanation: The value of pattern mask will be truncated in target column.

User response: Change the mask rule or field name in the mask rule definition. You may leave the mask rule untouched but note that its value will be truncated.

CKZ99065E SUBTASK *subtask_name*, INVALID MIE_RECfmt = *value_of_MIE_RECfmt*. 7 IS EXPECTED (_RFRI - REFERENTIAL INTEGRITY). RFIDSN.MIE_IDSNDNAME = *"data_set_name"* TABLE NAME = *"creator_name.table_name"*

Explanation: This is an internal error.

User response: Contact IBM Software Support.

CKZ99066E SUBTASK *subtask_number*, PARENT TABLE *"parent_creator_name.parent_table_name"* INVOLVED IN RELATION IS NOT FOUND RFIDSN.MIE_IDSNDNAME = *"data_set_name"* RI NAME = *"relation_name"* CHILD TABLE NAME = *"child_creator_name.child_table_name"*

Explanation: This is an internal error.

User response: Contact IBM Software Support.

CKZ99067E SUBTASK *subtask_name*, CHILD TABLE HAS RI FOR PARENT TABLE BUT PARENT TABLE HAS NOT RI FOR CHILD TABLE CHILD TABLE NAME = "*child_creator_name.child_table_name*" PARENT TABLE NAME = "*parent_creator_name.parent_table_name*" RI NAME = "*relation_name*" RFIDSN.MIE_IDSNAME = "*data_set_name*"

Explanation: This is an internal error.

User response: Contact IBM Software Support.

CKZ99068E SUBTASK *subtask_name*, RFKEY IS NOT FOUND IN PARENT RFRI CHILD TABLE NAME = "*child_creator_name.child_table_name*" PARENT TABLE NAME = "*parent_creator_name.parent_table_name*" RI NAME = "*relation_name*" RFIDSN.MIE_IDSNAME = "*data_set_name*"

Explanation: This is an internal error.

User response: Contact IBM Software Support.

CKZ99069E SUBTASK *subtask_number*, RFKEY IS NOT FOUND IN CHILD RFRI CHILD TABLE NAME = "*child_creator_name.child_table_name*" PARENT TABLE NAME = "*parent_creator_name.parent_table_name*" RI NAME = "*relation_name*" RFIDSN.MIE_IDSNAME = "*data_set_name*"

Explanation: This is an internal error.

User response: Contact IBM Software Support.

CKZ99070I SUBTASK *subtask_number*, CURRENT SUBTASK HAS BEEN INITIALIZED

Explanation: This message is informational.

User response: No action is required.

CKZ99071I SUBTASK *subtask_number*, CURRENT SUBTASK HAS BEEN TERMINATED

Explanation: This message is informational.

User response: No action is required.

CKZ99072I SUBTASK *subtask_number*, DATA MASKING ENGINE HAS BEEN TERMINATED SUCCESSFULLY

Explanation: This message is informational.

User response: No action is required.

CKZ99073W SUBTASK *subtask_number*, TERMINATION OF DATA MASKING ENGINE HAS COMPLETED WITH WARNINGS

Explanation: This message says that there were warnings during termination of data masking engine.

User response: See previous warning messages.

CKZ99074E SUBTASK *subtask_number*, UNABLE TO TERMINATE THE DATA MASKING ENGINE

Explanation: An error occurred during termination of data masking engine.

User response: See previous error messages for more information.

CKZ99075E SUBTASK *subtask_number*, FATAL ERROR - SEMGET() FAILED

Explanation: This is an internal error.

User response: Contact IBM Software Support.

CKZ99076I SUBTASK *subtask_number*, PRIVATE RANDOM GENERATOR SEMAPHORE HAS BEEN CREATED SUCCESSFULLY

Explanation: This message is informational.

User response: No action is required.

CKZ99077E SUBTASK *subtask_number*, FATAL ERROR - CAN'T INITIALIZE THE RANDOM GENERATOR SEMAPHORE

Explanation: This is an internal error.

User response: Contact IBM Software Support.

CKZ99078W SUBTASK *subtask_number*, CAN'T REMOVE THE RANDOM GENERATOR SEMAPHORE

Explanation: This is an internal warning. It does not affect the data producing by data masking.

User response: No action is required.

CKZ99079I SUBTASK *subtask_number*, RANDOM GENERATOR SEMAPHORE HAS BEEN REMOVED SUCCESSFULLY

Explanation: This message is informational.

User response: No action is required.

CKZ99080E SUBTASK *subtask_number*, FATAL ERROR - LOCKING OF RANDOM GENERATOR SEMAPHORE HAS FAILED

Explanation: This is an internal error.

User response: Contact IBM Software Support.

CKZ99081E SUBTASK *subtask_number*, FATAL ERROR - UNLOCKING OF RANDOM GENERATOR SEMAPHORE HAS FAILED

Explanation: This is an internal error.

User response: Contact IBM Software Support.

CKZ99082E Subtask *subtask_number*, Invalid range of parameter_name: [minimum_value, maximum_value]

Explanation: Invalid range of parameter in mask rule, i.e. minimum value exceeded the maximum value.

User response: See the next error messages to determine the incorrect mask rule and correct the range.

CKZ99083E Subtask *subtask_number*, (MIN_TIMESTAMPZ + MIN_TIMEZONE) is out of range

Explanation: Calculated minimum bound of TIMESTAMP WITH TIME ZONE is out of allowed range for this type of column in the RANDOM mask rule.

User response: Change MIN_TIMESTAMPZ and/or MIN_TIMEZONE parameters in the RANDOM mask rule, shown in the error messages following this error.

CKZ99084E Subtask *subtask_number*, (MAX_TIMESTAMPZ + MAX_TIMEZONE) is out of range

Explanation: Calculated maximum bound of TIMESTAMP WITH TIME ZONE is out of allowed range for this type of column in the RANDOM mask rule.

User response: Change MAX_TIMESTAMPZ and/or MAX_TIMEZONE parameters in the RANDOM mask rule shown in error messages following this error.

CKZ99087E Subtask *subtask_number*, iconv_open() error: Reason: *reason_code*; From CCSID: *from_CCSID*; To CCSID: *to_CCSID*

Explanation: String column data conversion error was issued. C function iconv_open() has failed. *reason_code* may be:

- EINVAL: the conversion between encoded character sets specified is not supported
- ECUNNOENV: a CUN_RS_NO_UNI_ENV error was issued by Unicode Conversion Services
- ECUNNOCONV: a CUN_RS_NO_CONVERSION error was issued by Unicode Conversion Services
- ECUNNOTALIGNED : a CUN_RS_TABLE_NOT_ALIGNED error was issued by Unicode Conversion Services
- ECUNERR: function iconv_open() encountered an unexpected error while using Unicode Conversion Services

User response: Contact IBM Software Support.

CKZ99088E Subtask *subtask_name*, iconv() error: Reason: *reason_code*; From CCSID: *from_CCSID*; To CCSID: *to_CCSID*

Explanation: String column data conversion error was issued. C function iconv() has failed. *reason_code* may be:

- EBADF: conversion descriptor is not valid
- ECUNNOENV: a CUN_RS_NO_UNI_ENV error was issued by Unicode Conversion Services
- ECUNNOCONV: a CUN_RS_NO_CONVERSION error was issued by Unicode Conversion Services
- ECUNNOTALIGNED: a CUN_RS_TABLE_NOT_ALIGNED error was issued by Unicode Conversion Services
- ECUNERR: function iconv() encountered an unexpected error while using Unicode Conversion Services
- EILSEQ: input conversion stopped due to an input byte that does not belong to the input codeset
- EINVAL: input conversion stopped due to an incomplete character or shift sequence at the end of the input buffer
- E2BIG: input conversion stopped due to lack of space in the output buffer

User response: Contact IBM Software Support.

CKZ99089E Subtask *subtask_number*, Unable to add rule '*mask_rule*' because CCSID is not defined for column *creator_name.table_name(column_name)*. Reason: *reason_code*

Explanation: CCSID is not defined for string column because it was created prior to Version 8. The data masking module attempted to retrieve the CCSID from SYSIBM.SYSTABLESPACE but encountered an error.

reason_code may be:

- 1: string column contains binary data and does not have valid CCSID
- 2: subtype of string column is not valid

- 3: database was created in a DB2 release prior to Version 5

User response: If *reason_code* = 1 then

1. Remove mask rule from list of mask rules
2. Change the mask rule to another column, or to the USEREXIT mask rule.

If *reason_code* = 2 or 3, contact IBM Software Support.

CKZ99090E Subtask *subtask_number*, Unable to add rule '*mask_rule*' because string column *creator_name.table_name(column_name)* contains binary data

Explanation: Unable to add defined mask rule to string column which contains binary data.

User response: Remove mask rule from list of mask rules, or change the mask rule to another column, or to the USEREXIT mask rule.

CKZ99091W Subtask *subtask_number*, Decimal floating-point value *decimal_floating_point_value_from_mask_rule* is out of range in *mask_rule_name* rule for column *creator_name.table_name(column_name)*. It is converted to *special_value*

Explanation: The defined decimal floating-point value is out of range. The value was converted to a *special_value*, which can be one of the following:

- +INFINITY
- -INFINITY
- +ZERO (+0E0)
- -ZERO (-0E0)

User response: To eliminate this message, change the decimal floating-point value in the mask rule so it is in the range of the current decimal floating-point format.

CKZ99092I Subtask *subtask_number*, *load_module_name build_date build_time* VERS=*version fff*

Explanation: This message is informational and provides the module level. The *fff* fields is currently unused.

User response: No action is required.

CKZ99107E MINIMAL (*minimum_parameter*) VALUE EXCEEDED OR EQUALED TO MAXIMAL (*maximum_parameter*) VALUE

Explanation: MINIMAL value exceeded or equaled to MAXIMAL value.

User response: Change MINIMAL or MAXIMAL value of range in the mask rule definition.

CKZ99109E UNABLE TO LOAD USER EXIT *user_exit_name*

Explanation: Unable to load user exit.

User response: Change user exit name in the mask rule definition.

CKZ99110E UNABLE TO PARSE SUPPLIED PATTERN '*pattern*'

Explanation: Unable to parse supplied mask pattern.

User response: See previous error messages for more information.

CKZ99117E EXPECTED SYMBOL '*symbol*' NOT RECEIVED

Explanation: Expected symbol not received in mask pattern. *sss* = symbol

User response: See error messages following this error for more information.

CKZ99118E INVALID MASK PATTERN '*mask_pattern*'

Explanation: Invalid mask pattern.

User response: See error messages following this error for more information.

CKZ99119E UNEXPECTED STRING '*string*' HAS BEEN RECEIVED

Explanation: An unexpected string has been received in mask pattern.

User response: See error messages following this error for more information.

CKZ99120E THE SPECIFIED STRING '*error_string*' DOES NOT CORRESPOND TO *mask_rule_name* PATTERN

Explanation: The specified string does not correspond to mask pattern.

User response: Correct the string in the mask rule definition.

CKZ99122E INVALID MASK RULE NAME: '*mask_rule_name*'

Explanation: Invalid mask rule name.

User response: Correct the mask rule name in the mask rule definition.

CKZ99182E Invalid range of *parameter_name*:
[*minimum_value*, *maximum_value*]

Explanation: Invalid range of parameter in mask rule; i.e. minimum value exceeded maximum value.

User response: See the next error messages to determine the incorrect mask rule and correct the range.

CKZ99183E (MIN_TIMESTAMPZ +
MIN_TIMEZONE) is out of range

Explanation: Calculated minimum bound of TIMESTAMP WITH TIME ZONE is out of the allowed range for this type of column in the RANDOM mask rule.

User response: Change MIN_TIMESTAMPZ and/or MIN_TIMEZONE parameters in the RANDOM mask rule shown in error messages following this error.

CKZ99184E (MAX_TIMESTAMPZ +
MAX_TIMEZONE) is out of range

Explanation: Calculated maximum bound of TIMESTAMP WITH TIME ZONE is out of allowed range for this type of column in the RANDOM mask rule.

User response: Change MAX_TIMESTAMPZ and/or MAX_TIMEZONE parameters in the RANDOM mask rule shown in error messages following this error.

CKZ99206I Unable to open DDNAME = *ddname* for
debug output.

Explanation: The DD name listed in the message could not be opened for debug output. Check accompanying z/OS messages for additional information.

User response: If unable to determine the reason for the failure, contact IBM Software Support. Have available the listing containing these messages.

CKZ99224E Unable to add unique key
unique_key(column_name).

Explanation: Insufficient memory was available to allocate the unique key listed in the message. Check accompanying z/OS messages for additional information.

User response: If unable to determine the reason for the failure, contact IBM Software Support. Have available the listing containing these messages.

CKZ99225E Unique key *unique_key(column_name)*
already exists).

Explanation: The name provided for the unique key already exists.

User response: Change the unique key name and regenerate the DDL.

CKZ99246I *module_name build_date build_time*
VERS=*version* REV=*revision* N/A
build_date.

Explanation: This message is used for troubleshooting purposes and provides information about the program and module.

User response: No action is required.

CKZDYN01I SYNTAX ERROR IN VALUE OF
XXXXXX -n | Illegal character in input
text | CKZ01PSN not available |
IEFDB476 not available | Unknown
request type | Function complete. RC=0
| DSN failed RACF test | Not enough
text units - | Critical keyword missing:
xxxx | SVC 99 (F) FAILURE. R15: xx
ERROR: xxxx INFO: xxxx

Explanation: Indicates CKZ01DYN failed during a dynamic allocation function.

User response: Report this error and associated product errors to IBM Software Support.

CKZERR00I OBJECT ACCESS FAILURE, DDNAME:
ddn

Explanation: These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response: Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please call IBM Software Support.

CKZERR01I VVDS ACCESS FAILURE, DDNAME:
ddn

Explanation: These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response: Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please call IBM Software Support.

CKZERR02I DSN: usercat name

Explanation: User Catalog cannot be accessed.

User response: Follow the Programmer Response for message IEC161I found in JESYSMSGs or eliminate the usercat from the selection list.

CKZERR03I NO DD ALLOCATION FOUND

Explanation: These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response: Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please call IBM Software Support.

**CKZERR04I FAILING module FUNCTION:
##,description**

Explanation: These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response: Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please call IBM Software Support.

**CKZERR05I PREVIOUS module FUNCTION:
##,description**

Explanation: These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response: Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended Support.

CKZERR06I PROGRAM CSECT: csect

Explanation: These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response: Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please call IBM Software Support.

CKZERR08I statement ERROR ID: listingID#

Explanation: These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response: Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please call IBM Software Support.

CKZERR09I statement ASM LISTING LINE #: line#

Explanation: These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response: Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please call IBM Software Support.

CKZERR10I R15 = ##, description

Explanation: These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response: Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please call IBM Software Support.

CKZERR12I module ASM LISTING LINE #: line

Explanation: These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response: Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please call IBM Software Support.

CKZERR14I PROCESSOR R15: r15

Explanation: These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response: Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please call IBM Software Support.

CKZERR15I PROCESSOR REASON CODE: nnn

Explanation: This is the decimal value of the failure reason code. Usually this is followed by message CKZERR20I containing text explanation of the reason code. If a text explanation is not available, CKZERR18I is issued, "NO DESCRIPTION FOR REASON CODE".

User response: If message CKZERR18I follows CKZERR15I and the error is determined by OPEN, CLOSE or VSAM (as indicated by message CKZERR19I), additional information about the reason code can be obtained by consulting IBM manual "z/OS Macro Instruction for Data Sets" section "VSAM Macro Return and Reason Codes". There are sub-sections for "OPEN", "CLOSE" and "Record Management" respectively.

CKZERR16I • CKZERR26I

CKZERR16I SVC99 ERROR CODE:

Explanation: These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response: Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please call IBM Software Support.

CKZERR17I SVC99 INFORMATION CODE:

Explanation: These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response: Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please call IBM Software Support.

CKZERR18I NO DESCRIPTION FOR REASON CODE

Explanation: These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response: Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please call IBM Software Support.

CKZERR19I ERROR DETERMINED BY:

Explanation: These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response: Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please call IBM Software Support.

CKZERR20I description

Explanation: These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response: Depends on other messages. Generally, these messages are designed to be self explanatory, and are used by support staff for extended debugging. Please call IBM Software Support.

CKZERR21I ** IMPLICIT OPEN

Explanation: These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response: Depends on other messages. Generally,

these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please call IBM Software Support.

CKZERR22I ** IMPLICIT CLOSE

Explanation: These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response: Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please call IBM Software Support.

CKZERR23I ** IMPLICIT REPOSITION

Explanation: These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response: Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please call IBM Software Support.

CKZERR24I OPEN CLASSIFICATION:

Explanation: These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response: Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please call IBM Software Support.

CKZERR25I CALLING PARM LRECL:

Explanation: These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response: Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please call IBM Software Support.

CKZERR26I CALLING PARM KEYLEN:

Explanation: These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response: Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please call IBM Software Support.

CKZERR27I LRECL:

Explanation: These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response: Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please call IBM Software Support.

CKZERR28I CI RBA REQUESTED:

Explanation: These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response: Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please call IBM Software Support.

CKZERR29I VVR/NVR KEY REQUESTED:

Explanation: These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response: Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please call IBM Software Support.

CKZERR30I RECORD TYPE REQUESTED:

Explanation: These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response: Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please call IBM Software Support.

CKZERR31I VVR KEYRANGE QUALIFIER REQUESTED:

Explanation: These messages are issued when the Catalog Fast Reader, Catalog Keyed Updater, or VVDS Handler encounter an error.

User response: Depends on other messages. Generally, these messages are designed to be self-explanatory, and are used by support staff for extended debugging. Please call IBM Software Support.

CKZG1010E The following space is not set to LOG for a required log range.

Explanation: The space listed in the messages is not set to LOG for a required log range.

User response: No action is required.

CKZG1012E No valid full image copy in the SYSCOPY history was found for space(s): spaces

Explanation: DB2 Cloning Tool was unable to find a valid full image copy in the SYSCOPY history for the table space(s) indicated in the message. DB2 Cloning Tool requires a full image copy registered in SYSCOPY.

User response: Ensure the image copy is registered in SYSCOPY and that it is valid.

CKZG1013E A table update ICTYPE was found in SYSCOPY that did not log for space(s): spaces

Explanation: There were multiple spaces being processed for which it was impossible for DB2 Cloning Tool to process due to the fact that some operation (such as LOAD REPLACE LOG(NO), REORG LOG(NO), etc.) occurred at some point between the selected starting point and the specified end point.

User response: No action is required.

CKZG1014I Database: database Space: space Partition: partition

Explanation: This message is issued in conjunction with other DB2 Cloning Tool messages to indicate the database, space, and partition for which other messages apply.

User response: No action is required.

CKZG1015E Could not determine disk/tape status of unit name.

Explanation: The device type for work data sets entered in the control file is invalid.

User response: Enter the correct device type.

CKZG1016E The device type of the unit name from the control file could not be determined.

Explanation: The device type for work data sets entered in the control file is invalid.

User response: Enter the correct device type.

CKZG1017E The REPORT utility returned an unrecoverable error.

Explanation: An internal error occurred.

User response: Contact IBM Software Support.

CKZG1018E The FULL image copy DD CA(LP/LB/RP/RB) {1} is missing from the JCL. Each CAxxxx DD correlates to each SPACE(...) control card group.

Explanation: The full image copy data set is not included in your DB2 Cloning Tool JCL.

User response: Verify that the JCL is formatted correctly and contains the necessary information for your DB2 Cloning Tool job.

CKZG1019E The FULL image copy DD CA {1} refers to a DSNAM already in SYSCOPY.

Explanation: You specified a full image copy data set name that already exists in SYSCOPY.

User response: Specify a different image copy data set name.

CKZG1020I Each CAxxxx DD correlates to each SPACE(...) control card group.

Explanation: Each CAxxxx DD statement must be associated with a corresponding SPACE(...) control card group.

User response: Verify that the JCL is formatted correctly and that each CAxxxx DD statement is associated with a SPACE(...) control card group.

CKZG1021E The TO_QUIESCE control card was specified, but no quiesce point was found.

Explanation: The TO_QUIESCE control card directs DB2 Cloning Tool to read the log and incorporate data into the image copy up to the most recent quiesce point but no quiesce point was found.

User response: No action is required.

CKZG1022E The stop point precedes the start point for space: Database: *database* Table space: *table_space* Partition: *partition* Start point *X'startpoint'* End point *X'endpoint'*.

Explanation: The DB2 Cloning Tool job will not run if the stop point proceeds the start point for the listed database, table space, partition.

User response: Correct the JCL and resubmit the job.

CKZG1023I The version of DB2 subsystem *ssid* is *ver*.

Explanation: Displays the SSID and the version of the DB2 subsystem.

User response: No action is required.

CKZG1024I The version of DB2 group attach *member_name* is *version*.

Explanation: Displays the version of DB2 group attach that the DB2 group attach member subsystem is running.

User response: No action is required.

CKZG1025I The table space *table_space* is non-partitioned.

Explanation: Indicates that the table space displayed in the message is non-partitioned.

User response: No action is required.

CKZG1027I DB2 Cloning Tool will process *dataset* for tablespace *tablespace*.

Explanation: Indicates the data set name that DB2 Cloning Tool will process.

User response: No action is required.

CKZG1028I The image copy is of all parts.

Explanation: Indicates that the image copy is of all partitions of the table space.

User response: No action is required.

CKZG1029I The image copy contains one partition (*partition*).

Explanation: Indicates the one partition that the image copy contains.

User response: No action is required.

CKZG1030E A concurrent image copy was found in the SYSCOPY history. It cannot be used.

Explanation: The DFDSS concurrent image copy that was found cannot be read by DB2 Cloning Tool.

User response: Select an alternative mechanism by which to recover the space.

CKZG1031I Only partition *partition* within the image copy will be updated with log data and written to an individual partition copy.

Explanation: DB2 Cloning Tool will only update the partition within the image copy with log data and will write to an individual partition image copy.

User response: No action is required.

CKZG1032I All partitions will be updated with log data.

Explanation: DB2 Cloning Tool will update all partitions with log data.

User response: No action is required.

CKZG1033E A partial recovery point was found in SYSCOPY and its data set name does not match the data set name specified in the STARTING_IC control card.

Explanation: Although you specified a particular starting point, it cannot be used because a partial recovery point was found in SYSCOPY and the data set associated with it has to be used instead.

User response: You must remove the STARTING_IC control card from your DB2 Cloning Tool syntax.

CKZG1034I DB2 Cloning Tool will process the log only for table space *table_space* PART *part*.

Explanation: DB2 Cloning Tool will process only the log for the indicated table space and partition.

User response: No action is required.

CKZG1035E An image copy was found, but its RBA precedes the logging start point.

Explanation: This message indicates that although an image copy was found, it could not be used since its RBA precedes the logging start point.

User response: No action is required.

CKZG1036I DB2 Cloning Tool processing ends.

Explanation: Indicates that DB2 Cloning Tool processing has completed.

User response: No action is required.

CKZG1038E An incremental image copy was marked as cataloged in SYSCOPY, but was not found in the MVS catalog.

Explanation: This message indicates that although an incremental image copy was marked as cataloged in SYSCOPY, it was not found in the MVS catalog.

User response: No action is required.

CKZG1039E At least two end points within a single GROUP() are not the same.

Explanation: At least two end points within a GROUP are not the same.

User response: Verify that the end points you defined are correct.

CKZG1040I The SPACE(...) set involved that the error was detected in was #'XXXXX'

Explanation: There was an error in the SPACE set indicated in the message.

User response: Verify the correct SPACE syntax has been specified.

CKZG1155I Control card stream processed by DB2 Cloning Tool follows...

Explanation: Indicates the control card stream that was processed by DB2 Cloning Tool.

User response: No action is required.

CKZG1156I DB2 Cloning Tool processing messages follow...

Explanation: Indicates that there are DB2 Cloning Tool messages that follow.

User response: Evaluate the message as necessary.

CKZG1200E The subsystem DB2 Cloning Tool was started with could not be found in JES2.

Explanation: The subsystem DB2 Cloning Tool was started with could not be found in JES2.

User response: Verify that you have specified the correct subsystem.

CKZG1201E The subsystem DB2 Cloning Tool was started with is not active in JES2.

Explanation: This message indicates that the subsystem that DB2 Cloning Tool was started with is not active in JES2.

User response: No action is required.

CKZG1202E There are no active DB2 members on this machine for this data sharing group.

Explanation: The data sharing group you specified does not have any active DB2 members so DB2 Cloning Tool processing cannot proceed.

User response: Specify a valid data sharing group attach name or a valid subsystem on which the DB2 Cloning Tool processing can run.

CKZG1203I DB2 subsystem is not defined to OS/390. Using group attach name instead.

Explanation: The DB2 subsystem you specified is not defined on OS/390®. If you use a group attach name, you will be able to connect to a DB2 subsystem that is active on OS/390.

User response: Edit your DB2 Cloning Tool setup to connect to a group attach name or to connect to a DB2 subsystem that is active on OS/390.

CKZG1204I DB2 subsystem is not active on OS/390. Using group attach name instead.

Explanation: The DB2 subsystem you specified is not active on OS/390. If you use a group attach name, you will be able to connect to a DB2 subsystem that is active on OS/390.

User response: Edit your DB2 Cloning Tool setup to connect to a group attach name or to connect to a DB2 subsystem that is active on OS/390.

CKZG1205I The subsystem DB2 Cloning Tool was started with is the group attach name.

Explanation: This message indicates the subsystem group attach name that DB2 Cloning Tool process is using.

User response: No action is required.

CKZG1206I The following subsystems are part of the data sharing group.

Explanation: This message, in conjunction with message CKZG1207I, provides the following information about the subsystem on which your DB2 Cloning Tool job ran:

- Subsystem—the subsystem.
- Member ID—the member ID.
- Defined to OS/390—whether this member is defined to OS/390.
- Active—whether this member is known to this OS/390 running on OS/390.

Note: DB2 Cloning Tool cannot detect the status of a member that is not running on this OS/390. Although a DB2 member may appear to be inactive, it may be running on another OS/390. Regardless, DB2 Cloning Tool reads the logs and processes all of the necessary files from each member of the data sharing group.

User response: No action is required.

CKZG1207I Subsystem: *subsystem* Member ID: *memberid* Defined to OS/390: *system* Active: *status*

Explanation: This message, in conjunction with message CKZG1206I, provides the following information about the subsystem on which your DB2 Cloning Tool job ran:

- The subsystem.
- The member ID.
- Whether or not this member is defined to OS/390.
- Whether or not this member is running on OS/390.

Note: DB2 Cloning Tool cannot detect the status of a member that is not running on this OS/390. Although a DB2 member may appear to be inactive, it may be running on another OS/390. Regardless, DB2 Cloning Tool reads the logs and processes all of the necessary files from each member of the data sharing group.

User response: No action is required.

CKZG1208I *ssids*

Explanation: This message displays the SSIDs that accompany messages CKZG1206I and CKZG1207I.

User response: No action is required.

CKZG1300I The ENQs for the spaces were successful.

Explanation: This message indicates that the ENQs for the table spaces completed successfully.

User response: No action is required.

CKZG1301E The ENQ for database *database* PART *part* was not successful.

Explanation: Indicates the database and partition for which the ENQs did not complete successfully.

User response: No action is required.

CKZG1402E The desired incremental image copy could not be allocated.

Explanation: DB2 Cloning Tool could not allocate the incremental image copy you specified.

User response: Verify that the file is not in use.

CKZG1416E DB2 Cloning Tool will process the following incremental image copy file(s):

Explanation: This message indicates the incremental image copy files that will be processed by DB2 Cloning Tool.

User response: No action is required.

CKZG1417E For table space: *table_space* PART *part*

Explanation: This message is issued in association with CKZG1416I and indicates the table space and partition to which CKZG1416I applies.

User response: No action is required.

CKZG1418I All start points are Sharelevel Reference; checkpoint processing skipped.

Explanation: This informational message indicates that checkpoint processing has been skipped since all start

points are Sharelevel Reference.

User response: No action is required.

CKZG1419E A mismatch between passed Zparm information and the JES SSCT was found.

Explanation: The ZPARM member for the source subsystem could not be found.

User response: Verify that the ZPARM information is accurate in the log apply job. Verify that the correct data sets containing the ZPARM member for the source subsystem are allocated. Contact IBM Software Support if the problem persists.

CKZG1420E An unexpected error occurred while trying to read the ZPARM information.

Explanation: The ZPARM member could not be found.

User response: Verify that the ZPARM information is accurate in the log apply job. Verify that the correct data sets containing the ZPARM member are allocated. Contact IBM Software Support if the problem persists.

CKZG1421E An unexpected error occurred while trying to read the bootstrap data set.

Explanation: The ZPARM member could not be found.

User response: Verify that the ZPARM information is accurate in the log apply job. Verify that the correct data sets containing the ZPARM member are allocated. Contact IBM Software Support if the problem persists.

CKZG1503I *data_set_name*

Explanation: This message accompanies CKZG1502I and indicates the log data set name that was last usable.

User response: No action is required.

CKZG1510I Error code #1:'code' #2: 'code'

Explanation: This message indicates an internal error that occurs when the log reader process cannot allocate an active or archive log file

User response: Diagnose the problem using the return codes listed in the message. Refer to *DB2 UDB for z/OS V8 Messages* (GC18-9602-01) and *DB2 UDB for z/OS V8 Codes* (GC18-9603-01) documentation to for more information.

CKZG1515I Log accumulated processing is beginning on subsystem ssid.

Explanation: Log accumulated processing has started on the subsystem indicated in the message.

User response: No action is required.

CKZG1516I Above the bar storage exhausted.

Explanation: The above the bar storage has been exhausted due to system or control card limits.

User response: The limit set by the maximum secondary allocation parameter has been met.

CKZG1518I Reading type dsn

Explanation: The message displays the type (archive or active) and data set name of the archive or active log data set being processed.

User response: No action is required.

CKZG1520I The log reader task #task_number finished.

Explanation: This message indicates that the processing of the log reader has completed.

User response: No action is required.

CKZG1521I Issuing HRECALL for log dataset dsname

Explanation: This is an informational message indicating that the log data set needed for processing has been recalled from migration.

User response: No action is required.

CKZG1606I The DB2 log record sort DD ddname was allocated.

Explanation: This message displays the input DDNAME. This message is output if DB2 Cloning Tool dynamically allocates the SORTIN2/SORTOUT2 DDNAMES. If the SORTIN2/SORTOUT2 DDNAMES are specified by the user in the step JCL, the messages does not appear and those DDs will be used as specified.

User response: No action is required.

CKZG1607I The DB2 log record sort DD ddname was allocated.

Explanation: This message displays the output DDNAME. This message is output if DB2 Cloning Tool dynamically allocates the SORTIN2/SORTOUT2 DDNAMES. If the SORTIN2/SORTOUT2 DDNAMES are specified by the user in the step JCL, the messages does not appear and those DDs will be used as specified.

User response: No action is required.

CKZG1805I Dynamic allocation return code=*rc*.

Explanation: This message indicates the dynamic allocation return code.

User response: No action is required.

CKZG1806I Image copy name=*image_copy_name*
RBA=*rba*

Explanation: Indicates the image copy name and RBA.

User response: No action is required.

CKZG1808I The full image copy *image_copy* could not be deallocated.

Explanation: DB2 Cloning Tool could not deallocate the full image copy.

User response: Verify that the file is not in use or damaged. Check with your systems administrator to ensure you have proper authorizations to access this file.

CKZG1900I Log range LRSN X'*lrsn*' to X'*lrsn*' is being processed.

Explanation: Indicates the log range that is being processed by DB2 Cloning Tool.

User response: No action is required.

CKZG1901I Log range RBA X'*rba*' to X'*rba*' is being processed.

Explanation: Indicates the log range that is being processed by DB2 Cloning Tool.

User response: No action is required.

CKZG2000E The output full image copy *image_copy* could not be opened.

Explanation: DB2 Cloning Tool could not open the output full image copy.

User response: Verify that the file is not in use and that you have the proper authority to access this file.

CKZG2001E The sorted incremental page file could not be opened.

Explanation: DB2 Cloning Tool could not open the sorted incremental page file.

User response: Verify that the file is not in use or damaged.

CKZG2002E The sorted log record file could not be opened.

Explanation: The sorted log file record was not available and could not be opened.

User response: Verify that the file is not in use and that you have the proper authority to access this file.

CKZG2003E An internal error occurred in the DB2 Cloning Tool merge section.

Explanation: An internal error occurred.

User response: Contact IBM Software Support.

CKZG2005E The number of pages in the incr. image copy is inconsistent with the page size.

Explanation: The page size you specified is not consistent with the number of pages in the incremental image copy.

User response: Correct the number of pages specified and resubmit the job.

CKZG2006E An unexpected EOF was encountered on the sorted log records file.

Explanation: An unexpected end of file was encountered.

User response: Contact IBM Software Support.

CKZG2007E The selected full IC has a DBID/PSID mismatch to the DB2 catalog.

Explanation: The DBID/PSID for the selected full image copy does not match those in the DB2 catalog.

User response: Correct the DBID/PSID for the selected full image copy.

CKZG2008I Number of pages read from the full image copy file(s)=*n*.

Explanation: This informational message indicate the number of pages that were read from the full image copy files.

User response: No action is required.

CKZG2009I Number of pages read from the incremental image copy file(s)=*n*.

Explanation: This informational message indicates the number of pages that were read from the incremental image copy data set(s).

User response: No action is required.

CKZG2010I Number of records read from the log apply file=*n*.

Explanation: This informational message indicates the number of pages that were read from the log apply file.

User response: No action is required.

CKZG2011I Number of pages written to the new full image copy file(s)=*n*.

Explanation: This informational message indicates the number of pages that were written to the new full image copy data set(s).

User response: No action is required.

CKZG2012I Number of pages written to the table/index space file(s)=*n*.

Explanation: Indicates the number of pages written to the table/index space files/

User response: No action is required.

CKZG2013I Since no changes were found for this data set, it has been deleted: *dsn*

Explanation: This message appears during dynamic allocation of an output image copy dataset and no output was written to that dataset. It is similar to the other message that is reported when no output is written to an output image copy data set in JCL.

User response: No action is required.

CKZG2014E Will proceed to be processed anyways due to control card FORCE_COPIES.

Explanation: This informational message indicates that DB2 Cloning Tool processing will continue due to the specification of the FORCE_COPIES control card.

User response: No action is required.

CKZG2015E A open failure occurred on the VSAM I/O module.

Explanation: An open failure occurred for the VSAM I/O module.

User response: Refer to message CKZG2023E for any dynamic allocation return codes and consult with your systems programmer. For information about the dynamic allocation return codes received, see the *MVS Programming Authorized Assembler Service Guide (SA22-7608)*.

CKZG2016E A close failure occurred on the VSAM I/O module.

Explanation: A close failure occurred for the VSAM I/O module.

User response: Contact IBM Software Support.

CKZG2017E A write failure occurred on the VSAM I/O module.

Explanation: A write failure occurred for the VSAM I/O module.

User response: Contact IBM Software Support.

CKZG2018E An open for update failure occurred on the VSAM I/O module.

Explanation: An open failure occurred for the VSAM I/O module.

User response: Contact IBM Software Support.

CKZG2019E A random fetch failure occurred on the VSAM I/OI module.

Explanation: A fetch failure occurred for the VSAM I/O module.

User response: Contact IBM Software Support.

CKZG2020E A random write failure occurred on the VSAM I/O module.

Explanation: A write failure occurred for the VSAM I/O module.

User response: Contact IBM Software Support.

CKZG2021E A random close failure occurred on the VSAM I/O module.

Explanation: A close failure occurred for the VSAM I/O module.

User response: Contact IBM Software Support.

CKZG2022E The underlying table/index space data set could not be found in MVS.

Explanation: The table/index space could not be found in MVS.

User response: No action is required.

CKZG2023E Dynamic allocation return code ='*return_code*'.

Explanation: This diagnostic message indicates data set allocation failure.

User response: Diagnose the problem using the return code. Refer to *DB2 UDB for z/OS V8 Messages*

(GC18-9602-01) and *DB2 UDB for z/OS V8 Codes* (GC18-9603-01) for more information.

CKZG2024I Object Database=*database* Space Name=*space_name* Partition=*partition* will have an image copy written anyway due to control card FORCE_COPIES.

Explanation: DB2 Cloning Tool will write an image copy for the object indicated in the message and override the WRITE_TO_VSAM control card, because the control card FORCE_COPIES has been specified with a value of Y.

User response: No action is required. If you do not want an image copy produced, specify FORCE_COPIES N.

CKZG2025E The SYSIN DD could not be allocated during WRITE_TO_VSAM processing.

Explanation: DB2 Cloning Tool was unable to allocate the SYSIN DD during WRITE_TO_VSAM processing.

User response: No action is required.

CKZG2026E The SYSIN DD could not be opened for output during WRITE_TO_VSAM processing.

Explanation: DB2 Cloning Tool was unable to open the SYSIN DD during WRITE_TO_VSAM processing.

User response: No action is required.

CKZG2027E Open error code =

Explanation: This message displays the open error code.

User response: No action is required.

CKZG2028E The SYSPRINT DD could not be allocated during WRITE_TO_VSAM processing.

Explanation: DB2 Cloning Tool was unable to allocate the SYSPRINT DD during WRITE_TO_VSAM processing.

User response: No action is required.

CKZG2029I Space *database.spacename* Part # *number* will be written to DSN.

Explanation: SWITCH_VCAT keyword in effect, data set name *dsn* was generated to place WRITE_TO_VSAM result.

User response: None.

CKZG2030E The attempt to recreate the underlying VSAM data set returned an error.

Explanation: DB2 Cloning Tool was unable to output to the VSAM file specified in your DB2 Cloning Tool job.

User response: Refer to message CKZG2023I for any dynamic allocation return codes and consult with your systems programmer. For information about the dynamic allocation return codes received, see the *MVS Programming Authorized Assembler Service Guide (SA22-7608)*.

CKZG2031E The LP image copy spanned tape could not be freed for a device switch.

Explanation: The dynamic allocation of the image copy data set to the spanned tape failed because the tape could not be freed for a device switch.

User response: Verify that the spanned tape is available for allocation.

CKZG2032E The initial LP image copy could not be allocated on the tape device.

Explanation: The allocation of the image copy data set to the tape device failed.

User response: Verify that the tape device is available for allocation.

CKZG2033E The LP image copy data set to be created on tape could not be opened.

Explanation: The image copy data set that is to be created cannot be opened.

User response: Verify that the image copy data set you specified in your DB2 Cloning Tool JCL is available for use and resubmit the DB2 Cloning Tool job.

CKZG2034E The LP image copy could not be allocated to the DASD device.

Explanation: The dynamic allocation of the data set to the DASD device failed.

User response: Verify that the device name is correct and that it is available for allocation.

CKZG2035E The LP image copy data set to be created on DASD could not be opened.

Explanation: The image copy data set that is to be created cannot be opened.

User response: Verify that the image copy data set you specified in your DB2 Cloning Tool JCL is available for use and resubmit the DB2 Cloning Tool job.

CKZG2036E The spanned LP image copy on tape could not be opened.

Explanation: The image copy data set that is to be created cannot be opened.

User response: Verify that the image copy data set you specified in your DB2 Cloning Tool JCL is available for use and resubmit the DB2 Cloning Tool job.

CKZG2037E The LB image copy spanned tape could not be freed for a device switch.

Explanation: The dynamic allocation of the image copy data set to the spanned tape failed because the tape could not be freed for a device switch.

User response: Verify that the spanned tape is available for allocation.

CKZG2038E The initial LB image copy could not be allocated onto the tape device.

Explanation: The allocation of the image copy data set to the tape device failed.

User response: Verify that the tape device is available for allocation.

CKZG2039E The LB image copy data set to be created on tape could not be opened.

Explanation: The image copy data set that is to be created cannot be opened.

User response: Verify that the image copy data set you specified in your DB2 Cloning Tool JCL is available for use and resubmit the DB2 Cloning Tool job.

CKZG2040E The LB image copy could not be allocated to the DASD device.

Explanation: The dynamic allocation of the data set to the DASD device failed.

User response: Verify that the device name is correct and that it is available for allocation.

CKZG2041E The LB image copy data set to be created on DASD could not be opened.

Explanation: The image copy data set that is to be created cannot be opened.

User response: Verify that the image copy data set you specified in your DB2 Cloning Tool JCL is available for use and resubmit the DB2 Cloning Tool job.

CKZG2042E The spanned LB image copy on tape could not be opened.

Explanation: The image copy data set that is to be created cannot be opened.

User response: Verify that the image copy data set you specified in your DB2 Cloning Tool JCL is available for use and resubmit the DB2 Cloning Tool job.

CKZG2043E The RP image copy spanned tape could not be freed for a device switch.

Explanation: The dynamic allocation of the image copy data set to the spanned tape failed because the tape could not be freed for a device switch.

User response: Verify that the spanned tape is available for allocation.

CKZG2044E The initial RP image copy could not be allocated onto the tape device.

Explanation: The allocation of the image copy data set to the tape device failed.

User response: Verify that the tape device is available for allocation.

CKZG2045E The RP image copy data set to be created on tape could not be opened.

Explanation: The image copy data set that is to be created cannot be opened.

User response: Verify that the image copy data set you specified in your DB2 Cloning Tool JCL is available for use and resubmit the DB2 Cloning Tool job.

CKZG2046E The RP image copy could not be allocated to the DASD device.

Explanation: The dynamic allocation of the data set to the DASD device failed.

User response: Verify that the device name is correct and that it is available for allocation.

CKZG2047E The RP image copy data set to be created on DASD could not be opened.

Explanation: The image copy data set that is to be created cannot be opened.

User response: Verify that the image copy data set you specified in your DB2 Cloning Tool JCL is available for use and resubmit the DB2 Cloning Tool job.

CKZG2048E The spanned RP image copy on tape could not be opened.

Explanation: The image copy data set that is to be created cannot be opened.

User response: Verify that the image copy data set you specified in your DB2 Cloning Tool JCL is available for use and resubmit the DB2 Cloning Tool job.

CKZG2049E The RB image copy spanned tape could not be freed for a device switch.

Explanation: The dynamic allocation of the image copy data set to the spanned tape failed because the tape could not be freed for a device switch.

User response: Verify that the spanned tape is available for allocation.

CKZG2050E The initial RB image copy could not be allocated onto the tape device.

Explanation: The allocation of the image copy data set to the tape device failed.

User response: Verify that the tape device is available for allocation.

CKZG2051E The RB image copy data set to be created on tape could not be opened.

Explanation: The image copy data set that is to be created cannot be opened.

User response: Verify that the image copy data set you specified in your DB2 Cloning Tool JCL is available for use and resubmit the DB2 Cloning Tool job.

CKZG2052E The RB image copy could not be allocated to the DASD device.

Explanation: The dynamic allocation of the data set to the DASD device failed.

User response: Verify that the device name is correct and that it is available for allocation.

CKZG2053E The RB image copy data set to be created on DASD could not be opened.

Explanation: The image copy data set that is to be created cannot be opened.

User response: Verify that the image copy data set you specified in your DB2 Cloning Tool JCL is available for use and resubmit the DB2 Cloning Tool job.

CKZG2054E The spanned RB image copy on tape could not be opened.

Explanation: The image copy data set that is to be created cannot be opened.

User response: Verify that the image copy data set you specified in your DB2 Cloning Tool JCL is available for use and resubmit the DB2 Cloning Tool job.

CKZG2055I A volume written to and left on the system could not be found.

Explanation: When DB2 Cloning Tool finishes writing to a tape data set, the tape cartridge is not rewound and ejected. It is left on the tape drive in case another data set needs to be written afterwards. Once any one data set is written, it is closed and code then goes back and reads internal MVS control blocks to get specifics about that dataset. If this subsequent code can't find the data set just written and closed, the error occurs.

User response: Contact IBM Software Support.

CKZG2057E The following mini log data set could not be deallocated from OS/390:

Explanation: The mini log data set could not be deallocated from OS/390 and could therefore not be used in DB2 Cloning Tool processing. This message is followed by CKZG2060I which displays the name of the mini log data set that could not be deallocated.

User response: Verify that the mini log data set is available for use.

CKZG2058E The following mini log data set could not be opened:

Explanation: The mini log data set could not be opened and could therefore not be used in CKZG processing. This message is followed by CKZG2060I which displays the name of the mini log data set that could not be opened.

User response: Verify that the mini log data set is available for use.

CKZG2059E The following mini log data set could not be allocated:

Explanation: The mini log data set could not be allocated and could therefore not be used in DB2 Cloning Tool processing. This message is followed by CKZG2060I which displays the name of the mini log data set that could not be allocated.

User response: Verify that the mini log data set is available for use.

CKZG2060E *dsname*

Explanation: This message displays a data set name that is associated with other messages.

User response: No action is required.

CKZG2061I The mini log file *filename* has been processed.

Explanation: The mini log indicated in the messages has been processed.

User response: No action is required.

CKZG2062I CKZG will attempt to use the MINI_LOG_DSN_2 data set instead.

Explanation: DB2 Cloning Tool was unable to use the MINI_LOG_DSN_1 data set so it will now attempt to use the MINI_LOG_DSN_2 data set.

User response: No action is required.

CKZG2063E A corrupted row was found in the mini log control table.

Explanation: DB2 Cloning Tool was unable to use the mini log control table due to a corrupted row.

User response: No action is required.

CKZG2064E Both mini log data sets for this space could not be opened.

Explanation: DB2 Cloning Tool attempted to open both mini log data sets for the space but was unable to do so.

User response: Verify that the mini log data sets are available for use.

CKZG2065I Number of records read from the merged mini log file(s)=

Explanation: The number of records displayed in this message were read from the merged mini log files.

User response: No action is required.

CKZG2066E An unexpected EOF was encountered on a merged mini log records file.

Explanation: DB2 Cloning Tool encountered an unexpected EOF on a merged mini log record file.

User response: No action is required.

CKZG2067E XLAT_DSN <DSN> will be used for <XLAT_TARGET>

Explanation: XLAT_DSN <DSN> was not found and there is no XLAT_VSAM or XLAT_COPY specified. XLAT_TARGET will be determined by format of DSN.

So if DSN conforms to DB2 space name format, new VSAM data set will be allocated, if not new sequential data set will be allocated.

User response: No action is required.

CKZG2068E The XML sequence number update process failed.

Explanation: Coordinating the internal XML sequence number during OBIDLAT processing could not be completed.

User response: Contact IBM Software Support.

CKZG2069E The space *space* resulted in the error condition.

Explanation: This message follows other error messages and identifies the space related to the errors.

User response: No action is required.

CKZG2069I The space *space* resulted in the error condition.

Explanation: Generic message that follows many other error messages.

User response: No action is required.

CKZG2070E The alternative SSID XML sequence column update program failed.

Explanation: Coordinating the internal XML sequence number during OBIDLAT processing could not be completed.

User response: Contact IBM Software Support.

CKZG2071E An XML update job is needed, but the XML output DSN is missing.

Explanation: The XML output DSN was not specified.

User response: Specify an XML output DSN.

CKZG2072E An XML update job is needed, but the XML output prefix is missing.

Explanation: The XML output prefix was not specified.

User response: Specify the XML output prefix.

CKZG2073E An XML update job is needed, but the XML template DSN is missing.

Explanation: The XML template DSN was not specified.

User response: Specify an XML template DSN.

CKZG2074E The XML template dataset could not be allocated.

Explanation: Unable to allocate the needed DSN.

User response: Make sure the DSN exists and is accessible.

CKZG2075E Control file loadlib information could not be obtained for *ssid*

Explanation: The control file is not up to date with this DB2 SSID.

User response: Update it via setup option 0.

CKZG2076E The XML template data set could not be opened.

Explanation: The data set was allocated but could not be opened.

User response: Contact IBM Software Support.

CKZG2077E The XML job output data set/member could not be allocated.

Explanation: The supplied data set could not be allocated.

User response: Make sure authority exists to allocate.

CKZG2078E The XML job output data set/member could not be opened.

Explanation: The data set was allocated but could not be opened.

User response: Check for proper access authority.

CKZG2079E The XML template does not conform to the automatically generated guidelines.

Explanation: The XML template generated by DB2 Cloning Tool has been altered to the point that it does not conform to expected design.

User response: Regenerate the XML template.

CKZG2080E The target SSID for XML translation is missing in the control cards.

Explanation: There is a missing parameter.

User response: Correct the JCL and resubmit the job.

CKZG2081I The SPACE(...) set involved that the error was detected in was *spacesetnumber*

Explanation: Generic message that follows many other messages.

User response: No action is required.

CKZG2082E The XML target SSID/DBname/TSname control cards are missing.

Explanation: Missing control cards in the Space(...) set.

User response: No action is required.

CKZG2083E The XML target SSID/DBname/TSname control cards are invalid.

Explanation: Syntax error in control cards.

User response: Correct the syntax.

CKZG2084I XML update job created for SSID=*'ssid'*.

Explanation: The job has been created.

User response: No action is required.

CKZG2085E No references to subsystem could not be found in the JES SSCT.

Explanation: The specified DB2 SSID is not defined to z/OS.

User response: Ensure that the name is correct or contact IBM Software Support.

CKZG2086E The sorted log file could not be allocated.

Explanation: An allocation error has occurred.

User response: Verify that the proper authorization is set.

CKZG2087E The sorted log file could not be opened.

Explanation: After allocating, could not open.

User response: Ensure proper authorization exists, or contact IBM Software Support.

CKZG2088E A log record page number exceeded the extent size boundary.

Explanation: A DB2 internal error occurred. The page number encoded into the log record points beyond the number of allowable pages for a DB2 extent.

User response: Send the dump and any table space / table creation details to IBM Software Support.

CKZG2091E The mini log data set *minilog_dsname* could not be allocated.

Explanation: The mini log data set could not be allocated and therefore could not be used in DB2 Cloning Tool processing.

User response: Verify that the mini log data set is available for use.

CKZG2092E The mini log data set *minilog_dsnname* could not be opened.

Explanation: The mini log data set could not be opened and therefore could not be used in DB2 Cloning Tool processing.

User response: Verify that the mini log data set is available for use.

CKZG2095I The sort of the applicable log records was successful.

Explanation: This message indicates that the sort of the applicable log records completed without error.

User response: No action is required.

CKZG2098E The attempt to delete the underlying VSAM data set returned an error.

Explanation: IDCAMS DELETE service returned an error.

User response: Check the IDCAMS output and correct the problem.

CKZG2099E Invalid image copy DSN:
image_copy_dsnname

Explanation: This message indicates that an error occurred when trying to process the image copy.

User response: Verify that the image copy specified in the message is available for use. If the problem persists, contact IBM Software Support.

CKZG2300E An internal error occurred unloading a mini log data set.

Explanation: An internal error occurred.

User response: Contact IBM Software Support.

CKZG2301E The following mini log data set could not be deallocated from OS/390:

Explanation: The specified mini log data set could not be deallocated from OS/390.

User response: Verify that you have specified the correct mini log data set name generation string.

CKZG2302E The following mini log data set could not be opened:

Explanation: The specified mini log data set could not be opened.

User response: Verify that the file is not in use and that you have the proper authority to access this file.

CKZG2303E The following mini log data set could not be allocated:

Explanation: The specified mini log data set could not be allocated.

User response: Verify that the file is not in use and that you have the proper authority to access this file.

CKZG2304I *dsn*

Explanation: Indicates the mini log DSN. This message is issued in conjunction with message CKZG2303I.

User response: No action is required.

CKZG2305I Dynamic allocation return code =
'return_code'

Explanation: This diagnostic message indicates data set allocation failure.

User response: Diagnose the problem using the return code. Refer to *DB2 UDB for z/OS V8 Messages* (GC18-9602-01) and *DB2 UDB for z/OS V8 Codes* (GC18-9603-01) for more information.

CKZG2310I The mini log file: *mini_log_file* has been processed.

Explanation: This message indicates the mini log file that has been processed.

User response: No action is required.

CKZG2312E A space level mini log DSN has the same name as a group level mini log DSN.

Explanation: Different groups of spaces in the log apply control card have the same mini log data set specified.

User response: IBM Software Support

CKZG2313E Mini log data set *dsn* could not be appended because a gap is found for the object in the mini log control table.

Explanation: There is a gap for the object in the mini log chain in the mini log control table. For this reason, the mini log data set indicated in the message could not be appended.

User response: To resolve this issue, either remove the mini log DSN from the mini log control table and MVS catalog or specify a new DSN for the mini log.

CKZG2401E The space *space* PART *part* has an unknown space status.

Explanation: This message ensures that the indicated space is to be stopped before proceeding with the WRITE_TO_VSAM process. DB2 Cloning Tool checks the space with a call similar to a '-display db(dbname) spacename(tsname) part(0)' to verify that the space is in 'stop' status. This message displays when the space comes back with a status not equal to RO, STOP, RW, or UT.

User response: Stop the indicated space before attempting to proceed with the WRITE_TO_VSAM process.

CKZG2402E The stop status check for space *space* PART *part* timed out.

Explanation: This message is output when DB2 Cloning Tool tries to start and it has to ensure that when doing WRITE_TO_VSAM processing that the spaces are indeed stopped. The stop step that is generated (prior to DB2 Cloning Tool) to do this sends commands to DB2 to stop the data sets, but it does not wait for the spaces to actually stop. If an in-flight URID is processing against the object and the stop is done, the space changes to 'STOPP' or stop pending until the URID finishes. It may also take DB2 some time to flush buffers. In either case, DB2 Cloning Tool does a check on the spaces before doing any real processing. If any of the spaces do not come back 'stop,' it waits a few seconds and checks again. After a few checks like this, it aborts, producing this message.

User response: Diagnose why the space will not stop.

CKZG2500E Fetching SYSIBM.SYSLOGRANGE data produced an error

Explanation: DB2 Cloning Tool encountered an error when attempting to fetch SYSIBM.SYSLOGRANGE data.

User response: No action is required. The report utility's output will be output after this message.

CKZG2501E REPORT utility text follows: *text*

Explanation: This message is the header line for the REPORT utility output that follows on the next line.

User response: No action is required.

CKZG2502I Skipping SYSIBM.SYSLGRNX processing.

Explanation: This informational messages indicates that DB2 Cloning Tool is not processing SYSIBM.SYSLGRNX because NO_SYSLGRNX was specified.

User response: No action is required.

CKZG2609I The LOG_COPY_PREFERENCE parameter was specified, but no value was found with it.

Explanation: Your JCL includes the LOG_COPY_PREFERENCE parameter but no value is specified with it.

User response: Specify a valid value for the LOG_COPY_PREFERENCE parameter.

CKZG2670I The MINILOG_SHARELEVEL keyword is ignored when not producing mini logs.

Explanation: You included the MINILOG_SHARELEVEL keyword in your JCL indicating the type of SHARELEVEL for mini logs but did not specify the production of producing mini logs. The MINILOG_SHARELEVEL keyword is therefore ignored.

User response: No action is required.

CKZG2672I The REPAIR_RECOVER_PENDING keyword is ignored when only writing to copies.

Explanation: The REPAIR_RECOVER_PENDING keyword is specified but this parameter is ignored when writing to copies.

User response: REPAIR_RECOVER_PENDING is ignored when writing to copies so the REPAIR_RECOVER_PENDING control card can be removed.

CKZG2703I Image copy name=*image_copy_name* RBA=*rba*.

Explanation: Indicates the image copy name an RBA.

User response: No action is required.

CKZG2706I DB2 Cloning Tool will process the following incremental image copy file(s):

Explanation: DB2 Cloning Tool will process the incremental image copy file(s) listed in this message.

User response: No action is required.

CKZG2707I For table space: *table_space* PART *part*

Explanation: This message indicates the table space and partition related to other DB2 Cloning Tool messages that have been issued.

User response: No action is required.

CKZG2804E An unexpected error occurred while trying to read the bootstrap data set.

Explanation: An unexpected error was encountered.

User response: Contact IBM Software Support.

CKZG2805E An unexpected error occurred while trying to read the ZPARM information.

Explanation: An unexpected error occurred.

User response: Contact IBM Software Support.

CKZG2806I The log apply process will begin at RBA=*'rba'*.

Explanation: The log apply process will start at the RBA indicated in the message.

User response: No action is required.

CKZG2812E A mismatch between passed Zparm information and the JES SSCT was found.

Explanation: This is an internal error indicating that the Zparm array that is being passed to DB2 Cloning Tool is inconsistent with the subsystem list found inside MVS' data structures.

User response: Contact IBM Software Support.

CKZG2813I The log reader process will launch a total of *nnnn* tasks.

Explanation: This message indicates the total number of tasks that will be launched.

User response: No action is required.

CKZG2814I The log reader will launch total of 1 task per member.

Explanation: Indicates that processing of the log reader will launch a total of one task per member since PARALELL has been set to 0.

User response: No action is required.

CKZG2815I The log reader process will start with PARALLEL tasks = *nnnn*

Explanation: The log reader process will start with the indicated maximum number of tasks.

User response: No action is required.

CKZG2816I The log reader task #*task_number* finished.

Explanation: Indicates that processing of the log reader finished.

User response: No action is required.

CKZG2817I The log reader task init failed.
RC=*return_code*

Explanation: The log apply processing failed to initialize a task necessary for reading logs. The reason code is specified in the error message.

User response: IBM Software Support

CKZG3000E The space *database.table_space* PART *partition* has an unknown space status.

Explanation: The status of the space indicated in the message is not known.

User response: When DB2 Cloning Tool checks the space to see if it is in recover pending, a status code unknown to DB2 Cloning Tool was found. Contact IBM Software Support.

CKZG3001E The stop status check for space *database.table_space* PART *partition* timed out.

Explanation: The stop status check for the space indicated in the message timed out.

User response: After the Repair operation is started, DB2 Cloning Tool checks the space afterwards, waiting for the recover pending flag to be removed by DB2. After checking 5 times in 15 seconds, the space was still in recover pending status. Remove the status manually.

CKZG3002E An attempt to Repair the Recover Pending status failed.

Explanation: The JCL specified to repair the recover pending status but the repair failed.

User response: When DB2 Cloning Tool called DB2 to repair the recover pending status for the space, the operation finished with an error condition. Contact IBM Software Support.

CKZG3003E An error occurred on an attempt to open the DSNUTILB Steplib.

Explanation: DB2 Cloning Tool was unable to open the DSNUTILB Steplib.

User response: The DB2 loadlib concatenation in the setup screens are not complete. When DSNUTILB attempted to use this concatenation, needed load modules were not found. Contact IBM Software Support.

CKZG3004E The Repair operation's SYSPRINT output dataset could not be opened.

Explanation: DB2 Cloning Tool was unable to open the repair operation's SYSPRINT output data set.

User response: Ensure the data set exists and is available for use.

CKZG3005E The Repair operation's SYSIN dataset allocation failed.

Explanation: DB2 Cloning Tool was unable to allocate the repair operation's SYSIN data set.

User response: To call DB2 to repair the recover pending status, a SYSIN data set needs to be allocated to hold the DB2 command stream. The data set could not be allocated. Check the setup screen work data set configuration for errors.

CKZG3006E Dynamic allocation return code = *return_code*

Explanation: This message indicates the return code associated with the failed dynamic allocation attempt.

User response: Ensure the data set exists and is available for use.

CKZG3007E The SYSIN DD could not be opened for output during Repair processing.

Explanation: DB2 Cloning Tool was unable to open the SYSIN DD during repair processing.

User response: After the SYSIN file is allocated, commands are written to it. This operation failed. Check the setup screen work data set configuration for errors.

CKZG3008E Open error code=*error_code*

Explanation: This message indicates the open error code encountered when attempting to open the SYSIN DD.

User response: No action is required.

CKZG3009E The Repair operation's SYSPRINT data set allocation failed.

Explanation: DB2 Cloning Tool was unable to allocate the repair operation's SYSPRINT data set.

User response: To call DB2 to repair the recover pending status, a SYSPRINT data set needs to be allocated to hold the DB2 command processor's output stream. The data set could not be allocated. Check the setup screen work data set configuration for errors.

CKZG3260I Incremental method SORT is obsolete. MERGE mode used instead.

Explanation: INCREMENTAL SORT is no longer supported (it is ignored). The internal method used instead is MERGE.

User response: None.

CKZG3401E The following XML SSID/DBname/TSname control card is invalid:

Explanation: The control cards do not conform to expected syntax.

User response: Ensure the generated job was not altered.

CKZG3402I *message_text*

Explanation: This message is generated with CKZG3401E.

User response: No action is required.

CKZG3405E Could not obtain SSID and User Indicator from input parameters.

Explanation: The log apply job was unsuccessful in trying to connect to the specified subsystem when processing spaces with XML data.

User response: Verify that the subsystem SSID specified in the job is accurate. Correct and resubmit the job. If the problem persists, contact IBM Software Support.

CKZG3406E Could not open the SYSOUT DD.

Explanation: The log apply job could not open the SYSOUT DD.

User response: Verify that the SYSOUT DD is specified in the job. Correct and resubmit the job. If the problem persists, contact IBM Software Support.

CKZG3407E Invalid SYSOUT DD LRECL.

Explanation: The LRECL specified on the SYSOUT DD is incorrect.

User response: Verify that the LRECL specified in the SYSOUT DD is accurate. Correct and resubmit the job. If the problem persists, contact IBM Software Support.

CKZG3450I Object *object* required no action.

Explanation: The object was determined to require no action to make the object usable.

User response: No action is required.

CKZG3451I Object object had its sequence nbr increased by *rowcount*.

Explanation: DB2 Cloning Tool updated the catalog to make the XML object usable.

User response: No action is required.

CKZG3452I With a source count=*count*

Explanation: DB2 Cloning Tool updated the catalog to make the XML object usable.

User response: No action is required.

CKZG3453E The XML sequence number could not be obtained for source object.

Explanation: Coordination of the internal XML sequence number during OBIDXLAT processing could not be completed.

User response: IBM Software Support

CKZG3500E The XML target SSID/DBname/TSname control cards are invalid.

Explanation: The subsystem, database name or table space name are invalid in the log apply control cards.

User response: Correct the subsystem, database name or table space name and resubmit the job. If the problem persists, contact IBM Software Support.

CKZG3501I The SPACE(...) set involved that the error was detected in was *spacesetnumber*

Explanation: Indicates the SPACE set involved in the error.

User response: No action is required.

CKZG3502E The XML sequence number could not be obtained for source object.

Explanation: Coordination of the internal XML sequence number during OBIDXLAT processing could not be completed.

User response: IBM Software Support

CKZG3604I The SPACE(...) set involved that the error was detected in was *spaceset*

Explanation: This error message is produced with other error messages and indicates the name of the space that is causing the error.

User response: Check other messages for processing errors.

CKZG3605E The end point for *database.table_space* did not match the UNIFIED value.

Explanation: The end point for the table space indicated in the message did not match the value specified for the UNIFIED value.

User response: No action is required.

CKZG3606E Consistency value = *X'consistency_token'*

Explanation: A problem occurred with the UNIFIED or UNIFIED_WARNING control cards. The message displays the consistency value taken from the first object in the group.

User response: The values shown in CKZG3606E and CKZG3607E can be compared for diagnostic purposes. The value shown in CKZG3606E is that for the first object in the group, while the value shown in CKZG3607E is the value for any object that does not match it. For example, if there are 10 objects in the group and three do not match the first, then one CKZG3606E message will display with three CKZG3607E messages (for each group).

CKZG3607E Object's derived value = *X'consistency_token'*

Explanation: A problem occurred with the UNIFIED or UNIFIED_WARNING control cards. The message displays the object's derived value for the first object in the group. This message displays any object that does not match the first object in the group (identified in CKZG3606E).

User response: The values shown in CKZG3606E and CKZG3607E can be compared for diagnostic purposes. The value shown in CKZG3606E is that for the first object in the group, while the value shown in CKZG3607E is the value for any object that does not match it. For example, if there are 10 objects in the group and three do not match the first, then one CKZG3606E message will display with three CKZG3607E messages (for each group).

CKZG3609E The resource optimization phase [1|2] of DB2 sort failed. RC=*'return_code'*

Explanation: DB2 Sort failed in the initialization step necessary for optimization.

User response: IBM Software Support

CKZG3611E There is not enough storage to perform the desired number of parallel sorts.

Explanation: There is not enough virtual storage space to perform the number of parallel sorts specified.

User response: Lower the number of parallel tasks specified or increase the amount of virtual memory specified available for the job, then resubmit the job.

CKZG3612E Insufficient total storage to perform the desired number of parallel sorts.

Explanation: The amount of storage available for a sort was insufficient.

User response: Increase your region size or reduce the number of parallel tasks, then resubmit the job.

CKZG3801E The conversion program returned an error.

Explanation: This message is the header line for additional messages that follow.

User response: No action is required.

CKZG3802E The DB2 Cloning Tool row conversion program ended unexpectedly.

Explanation: An error occurred in the DB2 Cloning Tool row conversion program.

User response: This message is accompanied by related messages that provide more information about the issue. If you cannot resolve the issue, note the job return code and contact IBM Software Support.

CKZG3830E An error occurred when processing CKZG exit call function: *function*

Explanation: This message indicates that an error occurred when processing the following function calls: Initialize, Open data set and Close data set.

User response: Check additional error messages. Contact IBM Software Support if the problem persists.

CKZG3831I Calling page exit function: *function*

Explanation: This informational message indicates that one of the following function calls are issued: Initialize, Open dataset and Close dataset.

User response: No action is required.

CKZINP01E ERROR PARSING PARMLIB INI# MEMBER ON LINE n | ERROR PARSING PARMLIB INI MEMBER ON LINE n

Explanation: An error was found in the CKZINI member. The line shows the current line being processed.

User response: Look for other messages and repair the CKZINI member.

CKZINP02E UNMATCHED */ | Continued statement at end of file

Explanation: An end-comment (*/) was found for which there was no begin-comment (/*).

User response: Remove the end-comment or insert a begin-comment in the appropriate location.

CKZINP05E TOKEN NAME LONGER THAN 72

Explanation: A token name longer than 72 bytes was found on the specified line.

User response: Reduce the length of the token to 72 or less.

CKZINP06E EXPECTED = AFTER TOKEN NAME

Explanation: An equal sign (=) was not found after the token name.

User response: Insert an equal sign (=) between the token and its value.

CKZINP07E EXPECTED = AFTER TOKEN NAME

Explanation: An equal sign (=) was not found after the token name.

User response: Insert an equal sign (=) between the token and its value.

CKZINP08E FOUND NON-NOTES TOKEN BEFORE 1ST SECTION NAME

Explanation: A token was found before any Section was specified. Only the "Notes" token is allowed before a section name. Reminder: Notes is provided for the customer and no product code can access the Notes value.

User response: Ensure that the first non-comment line and non-Notes token is a Section name.

CKZINP09E DUPLICATE SECTION/TOKEN FOUND | section | token

Explanation: A token cannot be defined twice for the same section.

User response: Remove the redundant token and retry.

CKZINP10E file function FAIL, RC = nnnnnnnn

Explanation: An unrecoverable error occurred during processing.

User response: Call IBM Software Support.

CKZINP11E Open failure, DD ddname | DD INI OPEN FAILURE | DD SYSPRINT open failure | DD UPDATE open failure | DD TRACEDD open failure

Explanation: An unrecoverable error occurred during processing.

User response: Call IBM Software Support.

CKZINP12E EXPECTED PARM VALUE OR RECORD, FOUND function

Explanation: An unrecoverable error occurred during processing.

User response: Call IBM Software Support.

CKZINP14E ERROR PARSING MSCINI

Explanation: MSCINI INIMERGE failed to parse the MSCINI input because it could not find the INIMERGE_SECTION values.

User response: Use the INI# member distributed by IBM and retry. If you need assistance, call IBM Software Support.

CKZINP15W COLUMNS 73-80 NOT BLANK ON LINE

Explanation: INI parser detected characters in columns 73-80. IBM has found that many INI errors occur because the person editing the INI does not see text in columns 73-80.

User response: Edit the PARMLIB INI member so that columns 73-80 are blank.

CKZINP21E SYSPLEX NAME MORE THAN 8 CHARACTERS

Explanation: The sysplex value on an INI section statement is invalid because it is too long.

User response: Repair the section name and retry.

CKZINP22E SYSTEM NAME MORE THAN 8 CHARACTERS

Explanation: The system value on an INI section statement is invalid because it is too long.

User response: Repair the section name and retry.

CKZINP31E SECTION NAME LONGER THAT 72

Explanation: The section name is limited in size.

User response: Repair the section name and retry.

CKZINP32E INVALID SECTION NAME

Explanation: The section name can have alphanumeric characters in addition to a period, dash, underscore or slash.

User response: Repair the section name and retry.

CKZINP33E PERIOD(".") FOUND IN INI# SECTION

Explanation: A period character in the section name is valid in a customer INI for denoting sysplex and system names, but is not valid in the distributed INI#.

User response: Repair the section name and retry.

CKZINP34E QUALIFIED INIMERGE_VALUES IN INI | QUALIFIED PRODUCT_INFO IN INI

Explanation: These sections cannot be qualified by sysplex and/or system name.

User response: Repair the section name and retry.

CKZINP35E TWO INIMERGE_VALUES SECTIONS FOUND

Explanation: The INIMERGE_VALUES may only appear once in the INI.

User response: Repair the section name and retry.

CKZINP36E QUALIFIED SECTION AFTER GENERIC

Explanation: A section was found with sysplex and/or system qualifications but it follows the same section without such qualification. This section can never be accessed and is invalid.

User response: Repair the section name and retry.

CKZINP37E DUPLICATE QUALIFIED SECTION | DUPLICATE UNQUALIFIED SECTION FOUND

Explanation: Two section names match exactly, then are either 1) both unqualified or 2) both specify the same Sysplex and system names. Section names must be unique.

User response: Repair the section name and retry.

CKZINP38E TRAILING PERIOD ON SECTION STATEMENT

Explanation: A section statement has one of the following formats:
:sectionname.sysplexname.systemname
:sectionname.sysplexname:sectionname..systemname

User response: Repair the section name and retry.

CKZINP39E INI# COLUMNS 73-80 ARE NOT BLANK, LINE nnnn | INI COLUMNS 73-80 ARE NOT BLANK, LINE nnnn

Explanation: Historically, placing text in columns 73-80 makes debugging errors in the INI difficult.

User response: Repair the section name and retry.

CKZINP40E RIGHT OF INI#'S SECTION NOT BLANK

Explanation: INI#'s section statement must be all blanks after section name to facilitate customers entering sysplex/system names.

User response: Repair the section name and retry.

CKZINP41I NOTHING COPIED

Explanation: The customer INI had no special sections to copy.

User response: No action is required.

CKZINP42I NOTHING DELETED

Explanation: No INI parameters were deleted.

User response: No action is required.

CKZINP50I INPUT: input line

Explanation: A previously noted error occurred on the line shown.

User response: See prior error message.

CKZINP51E TOKEN NON-NOTES BEFORE 1ST SECTION

Explanation: NOTES is the only token allowed before the first section statement.

User response: Repair the INI and resubmit.

CKZINP52E TOKEN NOTES FOUND IN INI#

Explanation: NOTES is only allowed in the customer INI.

User response: Repair the INI and resubmit.

CKZINP53E ERROR PARSING MSCINI (INI#), SEE JOB LOG | ERROR PARSING THE CUSTOMER INI, SEE JOB LOG

Explanation: INIMERGE used the INI parser to validate the INI but the INI parser was unsuccessful.

User response: For the customer INI, repair the INI based on the messages in the JOB log and resubmit.

For INI#, call IBM Software Support.

CKZINP71E SECTION IN INI# BUT NOT INI:

Explanation: INIVIEW cannot find section in INI that was found in the INI#.

User response: Repair the INI and resubmit.

CKZINP72I ALL INI# SECTIONS FOUND IN INI

Explanation: All of the INI# sections were found in the INI.

User response: No action is required.

CKZINP73E ERROR IN SYSPARM

Explanation: CKZ01INV SYSPARM is incorrect. Valid forms are: SYSPARM= for current SYSPLEX and SYS name SYSPARM=" for current SYSPLEX and SYS name SYSPARM=' for current SYSPLEX and SYS name SYSPARM=plexname,sysname'

User response: Correct the SYSPARM on your execute statement and resubmit.

CKZINS01E CKZ01VV1 END FAILURE, RC = nnnnnnnn

Explanation: An unrecoverable error occurred during processing.

User response: Call IBM Software Support.

CKZINS02E PASSED SECTION LENGTH INVALID length

Explanation: An unrecoverable error occurred during processing.

User response: Call IBM Software Support.

CKZINS03E PASSED TOKEN LENGTH INVALID length

Explanation: An unrecoverable error occurred during processing.

User response: Call IBM Software Support.

CKZINS04E INIGET 3RD PARM, NO VL BIT

Explanation: An unrecoverable error occurred during processing.

User response: Call IBM Software Support.

CKZINS05E INI TOKEN REC SHORTER THAN KEY SECTION/TOKEN section/token

Explanation: An unrecoverable error occurred during processing.

User response: Call IBM Software Support.

CKZINS06E TOKEN VALUE LONGER THAN RECEIVING PARM SECTION/TOKEN token TOKEN VALUE value

Explanation: An unrecoverable error occurred during processing.

User response: Call IBM Software Support.

CKZINS07E BAD REG 2 PTR TO VECTOR TABLE

Explanation: An unrecoverable error occurred during processing.

User response: Call IBM Software Support.

CKZINV00I CKZ01INV STARTING (REV=nn, PMR=xxxxxx, FIXDATE=xxxxxx)

Explanation: Message shows current version of the module.

User response: No action is required.

CKZINV01E ERROR PARSING PARMLIB INI# MEMBER ON LINE n | ERROR PARSING PARMLIB INI MEMBER ON LINE n

Explanation: An error was found in when parsing an INI member.

User response: Look for subsequent messages and repair the PARMLIB(xxxINI) or PARMLIB(xxxINI#) member as indicated.

CKZINV02E UMATCHED */ | CONTINUED STATEMENT AT END OF FILE

Explanation: An end-comment (*/) was found for which there was no begin-comment (/).

User response: Remove the end-comment or insert a begin-comment in the appropriate location.

CKZINV05E TOKEN NAME LONGER THAN 72

Explanation: A Token name longer than 72 bytes was found on the specified line.

User response: Reduce the length of the token to 72 or less.

CKZINV06E EXPECTED = AFTER TOKEN NAME

Explanation: An equal sign (=) was not found after the Token name.

User response: Insert an equal sign (=) between the Token and its value.

CKZINV07E EXPECTED = AFTER TOKEN NAME

Explanation: An equal sign (=) was not found after the Token name.

User response: Insert an equal sign (=) between the Token and its value.

CKZINV08E FOUND NON-NOTES TOKEN BEFORE 1ST SECTION NAME

Explanation: A Token was found before any Section was specified. Only the "Notes" Token is allowed before a section name. Reminder: Notes is provided for the customer and no product code can access the notes value.

User response: Ensure that the first non-comment line and non-Notes token is a Section name.

CKZINV09E DUPLICATE SECTION/TOKEN FOUND | Section | Token

Explanation: A token can not be defined twice for the same section.

User response: Remove the redundant token and retry.

CKZINV10E file function FAIL, RC = nnnnnnnn

Explanation: An unrecoverable error occurred during processing.

User response: Contact IBM Software Support.

CKZINV11E OPEN FAILURE, DD ddname | DD INI OPEN FAILURE | DD SYSPRINT OPEN FAILURE | DD UPDATE OPEN FAILURE | DD TRACEDD OPEN FAILURE

Explanation: An unrecoverable error occurred during processing.

User response: Contact IBM Software Support.

CKZINV12E EXPECTED PARM VALUE OR RECORD, FOUND function

Explanation: An unrecoverable error occurred during processing.

User response: Contact IBM Software Support.

CKZINV14E ERROR PARSING MSCINI

Explanation: MSCINI INIMERGE failed to parse the MSCINI input because it could not find the INIMERGE_SECTION values.

User response: Use the INI# member distributed by IBM Software Support and retry. If you need assistance, contact IBM Software Support.

CKZINV15W COLUMNS 73-80 NOT BLANK ON LINE

Explanation: INI parser detected characters in columns 73-80. IBM Software Support has found that many INI errors occur because the person editing the INI doesn't see text in columns 73-80.

User response: Edit the PARMLIB INI member so that columns 73-80 are blank.

CKZINV21E SYSPLEX NAME MORE THAN 8 CHARACTERS

Explanation: The sysplex value on an INI section statement is invalid because it is too long.

User response: Repair the section name and retry.

CKZINV22E SYSTEM NAME MORE THAN 8 CHARACTERS

Explanation: The system value on an INI section statement is invalid because it is too long.

User response: Repair the section name and retry.

CKZINV31E SECTION NAME LONGER THAT 72

Explanation: The section name is limited in size.

User response: Repair the section name and retry.

CKZINV32E INVALID SECTION NAME

Explanation: The section name can have alphanumeric characters in addition to a period, dash, underscore or slash.

User response: Repair the section name and retry.

CKZINV33E Period(".") found in INI# section

Explanation: A period character in the section name is valid in a customer INI for denoting sysplex and system names but is not valid in the distributed INI#.

User response: Repair the section name and retry.

CKZINV34E QUALIFIED INIMERGE_VALUES IN INI | QUALIFIED PRODUCT_INFO IN INI

Explanation: These sections can not qualified by sysplex and/or system name.

User response: Repair the section name and retry.

CKZINV35E TWO INIMERGE_VALUES SECTIONS FOUND

Explanation: The INIMERGE_VALUES may only appear once in the INI.

User response: Repair the section name and retry.

CKZINV36E QUALIFIED SECTION AFTER GENERIC

Explanation: A section was found with sysplex and/or system qualifications but it follows the same section without such qualification. This section can never be accessed and is invalid.

User response: Repair the section name and retry.

CKZINV37E DUPLICATE QUALIFIED SECTION | DUPLICATE UNQUALIFIED SECTION FOUND

Explanation: Two section names match exactly, then are either 1) both unqualified or 2) both specify the same sysplex and system names. Section names must be unique.

User response: Repair the section name and retry.

CKZINV38E TRAILING PERIOD ON SECTION STATEMENT

Explanation: A section statement has one of the following formats:

:sectionname.sysplexname.systemname

:sectionname.sysplexname :sectionname..systemname

User response: Repair the section name and retry.

CKZINV39E INI# COLUMNS 73-80 ARE NOT BLANK, LINE nnnn | INI COLUMNS 73-80 ARE NOT BLANK, LINE nnnn

Explanation: Historically, placing text in columns 73-80 makes debugging errors in the INI difficult.

User response: Repair the section name and retry.

CKZINV40E RIGHT OF INI#"S SECTION NOT BLANK

Explanation: INI#'s section statement must be all blanks after section name to facilitate customers entering sysplex/system names.

User response: Repair the section name and retry.

CKZINV41I NOTHING COPIED

Explanation: The customer INI had no special sections to copy

User response: No action is required.

CKZINV42I NOTHING DELETED

Explanation: No INI parameters were deleted.

User response: No action is required.

CKZINV50I INPUT: input line

Explanation: A previously noted error occurred on the line shown.

User response: See prior error message.

CKZINV51E TOKEN NON-NOTES BEFORE 1ST SECTION

Explanation: NOTES is the only token allowed before the first section statement.

User response: Repair the INI and resubmit.

CKZINV52E TOKEN NOTES FOUND IN INI#

Explanation: NOTES is only allowed in the customer INI.

User response: Repair the INI and resubmit.

CKZINV53E ERROR PARSING MSCINI (INI#), SEE JOB LOG | ERROR PARSING THE CUSTOMER INI, SEE JOB LOG

Explanation: INIMERGE used the INI parser to validate the INI but the INI parser was unsuccessful.

User response: For INI#, contact IBM Software Support.

For the customer INI, repair the INI based on the messages in the JOB log and resubmit.

CKZINV71E SECTION IN INI# BUT NOT INI:

Explanation: INIVIEW cannot find section in INI that was found in the INI#.

User response: Repair the INI and resubmit.

CKZINV72I ALL INI# SECTIONS FOUND IN INI

Explanation: All of the INI# sections were found in the INI.

User response: No action is required.

CKZINV73E ERROR IN SYSPARM

Explanation: CKZ01INV SYSPARM is incorrect. Valid forms are: SYSPARM= for current SYSPLEX and SYS name SYSPARM=" for current SYSPLEX and SYS name SYSPARM=' for current SYSPLEX and SYS name SYSPARM=plexname,sysname'

User response: Correct the SYSPARM on your execute statement and resubmit.

CKZMER01E ERROR PARSING PARMLIB INI# MEMBER ON LINE n | ERROR PARSING PARMLIB INI MEMBER ON LINE n

Explanation: An error was found in the CKZINI member. The line shows the current line being processed.

User response: Look for other messages and repair the CKZINI member.

CKZMER02E UNMATCHED */ | Continued statement at end of file

Explanation: An end-comment (*/) was found for which there was no begin-comment (/).

User response: Remove the end-comment or insert a begin-comment in the appropriate location.

CKZMER05E TOKEN NAME LONGER THAN 72

Explanation: A token name longer than 72 bytes was found on the specified line.

User response: Reduce the length of the token to 72 or less.

CKZMER06E EXPECTED = AFTER TOKEN NAME

Explanation: An equal sign (=) was not found after the token name.

User response: Insert an equal sign (=) between the token and its value.

CKZMER07E EXPECTED = AFTER TOKEN NAME

Explanation: An equal sign (=) was not found after the token name.

User response: Insert an equal sign (=) between the token and its value.

CKZMER08E FOUND NON-NOTES TOKEN BEFORE 1ST SECTION NAME

Explanation: A token was found before any Section was specified. Only the "Notes" token is allowed before a section name. Reminder: Notes is provided for the customer and no product code can access the Notes value.

User response: Ensure that the first non-comment line and non-Notes token is a Section name.

CKZMER09E DUPLICATE SECTION/TOKEN FOUND | section | token

Explanation: A token cannot be defined twice for the same section.

CKZMER10E • CKZMER37E

User response: Remove the redundant token and retry.

CKZMER10E file function FAIL, RC = nnnnnnnn

Explanation: An unrecoverable error occurred during processing.

User response: Call IBM Software Support.

CKZMER11E DD INI OPEN FAILURE | DD
SYSPRINT open failure | DD UPDATE
open failure | DD TRACEDD open
failure

Explanation: An unrecoverable error occurred during processing.

User response: Call IBM Software Support.

CKZMER12E EXPECTED PARM VALUE OR
RECORD, FOUND function

Explanation: An unrecoverable error occurred during processing.

User response: Call IBM Software Support.

CKZMER14E ERROR PARSING MSCINI

Explanation: MSCINI INIMERGE failed to parse the MSCINI input because it could not find the INIMERGE_SECTION values.

User response: Use the INI# member distributed by IBM and retry. If you need assistance, call IBM Software Support.

CKZMER15W COLUMNS 73-80 NOT BLANK ON
LINE

Explanation: INI parser detected characters in columns 73-80. IBM has found that many INI errors occur because the person editing the INI does not see text in columns 73-80.

User response: Edit the PARMLIB INI member so that columns 73-80 are blank.

CKZMER21E SYSPLEX NAME MORE THAN 8
CHARACTERS

Explanation: The sysplex value on an INI section statement is invalid because it is too long.

User response: Repair the section name and retry.

CKZMER22E SYSTEM NAME MORE THAN 8
CHARACTERS

Explanation: The system value on an INI section statement is invalid because it is too long.

User response: Repair the section name and retry.

CKZMER31E SECTION NAME LONGER THAN 72

Explanation: The section name is limited in size.

User response: Repair the section name and retry.

CKZMER32E INVALID SECTION NAME

Explanation: The section name can have alphanumeric characters in addition to a period, dash, underscore or slash.

User response: Repair the section name and retry.

CKZMER33E PERIOD(".") FOUND IN INI#
SECTION

Explanation: A period character in the section name is valid in a customer INI for denoting sysplex and system names, but is not valid in the distributed INI#.

User response: Repair the section name and retry.

CKZMER34E QUALIFIED INIMERGE_VALUES IN
INI | QUALIFIED PRODUCT_INFO IN
INI

Explanation: These sections cannot be qualified by sysplex and/or system name.

User response: Repair the section name and retry.

CKZMER35E TWO INIMERGE_VALUES SECTIONS
FOUND

Explanation: The INIMERGE_VALUES may only appear once in the INI.

User response: Repair the section name and retry.

CKZMER36E QUALIFIED SECTION AFTER
GENERIC

Explanation: A section was found with sysplex and/or system qualifications but it follows the same section without such qualification. This section can never be accessed and is invalid.

User response: Repair the section name and retry.

CKZMER37E DUPLICATE QUALIFIED SECTION |
DUPLICATE UNQUALIFIED SECTION
FOUND

Explanation: Two section names match exactly, then are either 1) both unqualified or 2) both specify the same Sysplex and system names. Section names must be unique.

User response: Repair the section name and retry.

**CKZMER37E DUPLICATE QUALIFIED SECTION |
DUPLICATE UNQUALIFIED SECTION
FOUND**

Explanation: Two section names match exactly, then are either 1) both unqualified or 2) both specify the same Sysplex and system names. Section names must be unique.

User response: Repair the section name and retry.

**CKZMER38E TRAILING PERIOD ON SECTION
STATEMENT**

Explanation: A section statement has one of the following formats:
:sectionname.sysplexname.systemname
:sectionname.sysplexname :sectionname..systemname

User response: Repair the section name and retry.

**CKZMER39E INI# COLUMNS 73-80 ARE NOT
BLANK, LINE nnnn**

Explanation: Historically, placing text in columns 73-80 makes debugging errors in the INI difficult.

User response: Repair the section name and retry.

**CKZMER40E RIGHT OF INI#'S SECTION NOT
BLANK**

Explanation: INI#'s section statement must be all blanks after section name to facilitate customers entering sysplex/system names.

User response: Repair the section name and retry.

CKZMER41I NOTHING COPIED

Explanation: The customer INI had no special sections to copy.

User response: No action is required.

CKZMER42I NOTHING DELETED

Explanation: No INI parameters were deleted.

User response: No action is required.

CKZMER50I INPUT: input line

Explanation: A previously noted error occurred on the line shown.

User response: See prior error message.

**CKZMER51E TOKEN NON-NOTES BEFORE 1ST
SECTION**

Explanation: NOTES is the only token allowed before the first section statement.

User response: Repair the INI and resubmit.

CKZMER52E TOKEN NOTES FOUND IN INI#

Explanation: NOTES is only allowed in the customer INI.

User response: Repair the INI and resubmit.

**CKZMER53E ERROR PARSING MSCINI (INI#),
SEE JOB LOG | ERROR PARSING
THE CUSTOMER INI, SEE JOB LOG**

Explanation: INIMERGE used the INI parser to validate the INI but the INI parser was unsuccessful.

User response: For INI#, call IBM Software Support.

For the customer INI, repair the INI based on the messages in the JOB log and resubmit.

CKZMER71E SECTION IN INI# BUT NOT INI:

Explanation: INIVIEW cannot find section in INI that was found in the INI#.

User response: Repair the INI and resubmit.

CKZMER72I ALL INI# SECTIONS FOUND IN INI

Explanation: All of the INI# sections were found in the INI.

User response: No action is required.

CKZMER73E ERROR IN SYSPARM

Explanation: CKZ01INV SYSPARM is incorrect. Valid forms are: SYSPARM= for current SYSPLEX and SYS name SYSPARM=" for current SYSPLEX and SYS name SYSPARM=' for current SYSPLEX and SYS name SYSPARM='plexname,sysname'

User response: Correct the SYSPARM on your execute statement and resubmit.

CKZMOD01I Starting (rev=xxx,ptf=xxx,fixdate=xxx)

Explanation: Indicates CKZ01MOD is starting and its version.

User response: Informational only

CKZMOD02I Terminating

Explanation: Indicates CKZ01MOD is terminating.

User response: Informational only

CKZMOD02E DIFF option requires SYSUT2

Explanation: DIFF compares loadlibs SYSUT1 and SYSUT2.

User response: Correct JCL or execution parameter

CKZMOD03E Unable to open SYSUT1

Explanation: SYSUT1 is a required DD statement.

User response: Correct JCL or execution parameter

CKZMOD04E A duplicate PMR found module

Explanation: The same PMR number was used twice in the same module.

User response: Informational only

**CKZMOD05E No PMR data available for 'SYSUT1
xxxxxxx' | No PMR data available for
'SYSUT2 xxxxxxx'**

Explanation: The load module has an EHDR, not MSCHDR/MSCPMR macros.

User response: Informational only

**CKZMOD06E No Header data available for 'SYSUT1
xxxxxxx' | No Header data available for
'SYSUT2 xxxxxxx'**

Explanation: The load module does not have MSCHDR/MSCPMR or EHDR macros.

User response: Informational only

**CKZMOD30E FATAL ERROR IN CKZ01VV1
TABLE(table) FUNC(func) RC(rc)**

Explanation: An error occurred using an CKZ01VV1 table.

User response: Please report this message to IBM Software Support.

**CKZMOD91E Error READ_PDSE ERROR -
terminating**

Explanation: An error occurred reading a PDSE (not a PDS)

User response: Please report this message to IBM Software Support.

**CKZMOD92E Error READ_PDS ERROR -
terminating**

Explanation: An error occurred reading a PDS (not a PDSE)

User response: Please report this message to IBM Software Support.

CKZPP100I text

Explanation: Message produced when command input is read.

User response: No action is required.

CKZPP101E Input file not open

Explanation: Message produced when command input is parsed.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZPP102E Input file LRECL invalid

Explanation: Message produced when command input is parsed. This error typically occurs if SYSIN DD statement refers to a data set with LRECL other than 80.

User response: Correct the command input and resubmit.

CKZPP103E Blank record invalid here

Explanation: Message produced when command input is parsed.

User response: Correct the command input and resubmit.

CKZPP104E Expected continuation not found

Explanation: Message produced when command input is parsed.

User response: Correct the command input and resubmit.

CKZPP105W Input flushed

Explanation: Message produced when command input is parsed and a prior error has been detected.

User response: Review the prior CKZPP1 error messages. Correct the command input and resubmit.

CKZPP106E Unmatched parenthesis

Explanation: Message produced when command input is parsed.

User response: Correct the command input and resubmit.

CKZPP107E Paren nesting error

Explanation: Message produced when command input is parsed.

User response: Correct the command input and resubmit.

CKZPP108E Line buffer full

Explanation: Message produced when command input is parsed.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZPP109E String delimiters not balanced

Explanation: Message produced when command input is parsed. Second quotation character not found.

User response: Correct the command input and resubmit.

CKZPP110E Input record buffer length invalid

Explanation: Message produced when command input is parsed.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZPSE20I Command ended

Explanation: Message produced when command input is parsed and a prior error has been detected.

User response: Review the prior CKZPSE error messages. Correct the command input and resubmit.

CKZPSE21E Keyword not found : text

Explanation: Message produced when command input is parsed. The displayed keyword is not valid for this command or parent keyword.

User response: Review the prior CKZPSE error messages. Correct the command input and resubmit.

CKZPSE22E Module not found: module-name

Explanation: Message produced when command input is parsed.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZPSE23E Operand not supported for: text

Explanation: Message produced when command input is parsed. The displayed keyword is not implemented.

User response: Correct the command input and resubmit.

CKZPSE24E Operand truncated for:

Explanation: Message produced when command input is parsed. The value specified for the displayed keyword is longer than the keyword accepts.

User response: Correct the command input and resubmit.

CKZPSE25E Empty or missing parmlist

Explanation: Message produced when command input is parsed and a prior error has been detected.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZPSE26W Command abandoned

Explanation: Message produced when command input is parsed and a prior error has been detected.

User response: Review the prior CKZPSE error messages. Correct the command input and resubmit.

**CKZPSE28E Operand required for keyword:
keyword**

Explanation: Message produced when command input is parsed. The displayed keyword requires a value.

User response: Contact IBM Software Support. Have available the listing that contains this message.

**CKZPSE29E Conflicting keywords specified with
keyword:**

Explanation: Message produced when command input is parsed. Two mutually exclusive keywords have been specified.

User response: Correct the command input and resubmit.

CKZPSE30E Multiple use of keyword not allowed

Explanation: Message produced when command input is parsed. Two mutually exclusive keywords have been specified.

User response: Correct the command input and resubmit.

**CKZRX101E CKZRNTGT ddname READ FAILED,
RC IS nn**

Explanation: The Rexx program was unable to read a required file.

User response: Add the appropriate DD statement to the execution JCL.

**CKZRX102E CKZRNTGT COUNTS DO NOT
MATCH: CKZIN = nnn; NUCIN = nnn**

Explanation: There is an unequal number of records in the file from DB2 Cloning Tool and the file from DB2 Cloning Tool NOUSERCATALOGS.

User response: Verify that the input files correspond

CKZRX103I • CKZRX322E

to the correct runs for the two COPYs.

CKZRX103I CKZRNTGT ddname HEADER IS header information

Explanation: Informational message.

User response: No action is required.

CKZRX104I CKZRNTGT ddname INPUT FILE HAS nnn VOLUME PAIRS

Explanation: Informational message.

User response: No action is required.

CKZRX105E CKZRNTGT NEWTGT OPEN FAILED, RC IS nn

Explanation: The REXX program was unable to open a required file.

User response: Add the appropriate DD statement to the execution JCL.

CKZRX106E CKZRNTGT NEWTGT WRITE FAILED, RC IS nn

Explanation: The REXX program was unable to add a record to the file.

User response: Check that the DCB attributes for the output file are correct. Check that there is sufficient space given to the output file.

CKZRX107E CKZRNTGT NO MATCH FOUND FOR CKZIN ENTRY source target

Explanation: No match was found for the indicated DB2 Cloning Tool source and target volume serials in the DB2 Cloning Tool NOUSERCATALOGS input file.

User response: Verify that the input files correspond to the correct runs for the two COPYs.

CKZRX108E CKZRNTGT VERSION MISMATCH: CKZIN: version NUCIN: version

Explanation: The version entries in the two files do not match.

User response: Verify that the input files correspond to the correct runs for the two COPYs.

CKZRX109E CKZRNTGT VERSION IS INCORRECT, version

Explanation: The version entry in the input files is not supported.

User response: Verify that the input files correspond to the correct runs for the two COPYs.

CKZRX110E CKZRNTGT CKZIN PRODUCT IS INCORRECT, product

Explanation: The product entry in the DB2 Cloning Tool input file is incorrect.

User response: Verify that the DB2 Cloning Tool input file has not been modified.

CKZRX111E CKZRNTGT NUCIN PRODUCT IS INCORRECT, product

Explanation: The product entry in the DB2 Cloning Tool NOUSERCATALOGS input file is incorrect.

User response: Verify that the DB2 Cloning Tool NOUSERCATALOGS input file has not been modified.

CKZRX112E CKZRNTGT NO INPUT READ

Explanation: No records were found in the input files.

User response: Verify that the input files correspond to the correct runs for the two COPYs and that the DD statements have not been dummied.

| **CKZRX306E modid Write to CMDOUT failed; return code = return_code**

| **Explanation:** The REXX program was unable to write to the output file allocated to the CMDOUT DD statement. Processing terminates.

| **User response:** Ensure that the DCB attributes for the CMDOUT file are correct. Ensure that there is sufficient space that is given to the output file. If unable to resolve the problem, contact IBM Software Support. Have available the listing that contains this message.

| **CKZRX321I modid Using DB2 SSID= ssid**

| **Explanation:** The REXX program will use the indicated DB2 ssid. The ssid value is obtained from the argument string that is provided to the REXX program.

| **User response:** No action is required.

| **CKZRX322E modid DSN command failed; return code= return_code**

| **Explanation:** The invocation of the DB2 DSN command processor failed. *return_code* is the return code from the invocation.

| **User response:** Verify that the correct DB2 ssid is being used, that the job is running on the same system as the DB2 system, and that the correct STEPLIB is being used. If unable to resolve problem, contact IBM Software Support. Have available the listing that contains this message.

CKZRX323I *modid* Number of commands written to
CMDOUT: *number_of_commands*

Explanation: The number of start commands that were written to CMDOUT DD statement.

User response: No action is required.

CKZRX324I *modid* There are no objects with UT status

Explanation: There are no DB2 objects found that have UT status, therefore no start commands need to be generated.

User response: No action is required.

CKZRX325I *modid* Start commands generated for:

Explanation: This message precedes a list of the *database.spacename* objects that have UT status, and that start commands have been generated for.

User response: No action is required.

CKZRX399E *modid* Exec error condition has occurred:
error_data

Explanation: The REXX program had an internal error. *error_data* identifies the statement in error. Processing terminates.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZTCZ01E *modid* Discover exec error condition has occurred: *error_information*

Explanation: An error occurred during the execution of the Discover EXEC. Information about the error is also displayed.

User response: If unable to determine the reason for the failure from the *error_information*, contact IBM Software Support. Have available a copy of the message and the error information.

CKZTCZ02E *modid* Discover EXEC invoked with no parameters.

Explanation: The Discover EXEC was invoked with no parameters.

User response: Ensure the Discover EXEC was executed from the Tools Customizer Discover Customized Product Information panel. If unable to determine the reason for the failure, contact IBM Software Support.

CKZTCZ03E *modid* Discover EXEC found empty CKZINI member.

Explanation: The Discover EXEC found no records in the CKZINI member.

User response: Ensure the correct data set name was specified and that the data set contains a valid CKZINI member.

CKZTCZ04E *modid* Data set *data_set_name* has no CKZINI member.

Explanation: The indicated data set does not have a CKZINI member.

User response: Ensure the correct data set name was specified and that the data set contains a valid CKZINI member.

CKZTCZ05E *modid* Data set *data_set_name* not usable for Discover: *reason*

Explanation: The indicated data set is not usable because of the indicated reason.

User response: Ensure the correct data set name was specified.

CKZVE201E NO INIT FUNCTION RECEIVED.

Explanation: This is an internal error.

User response: Call IBM Software Support.

CKZVE202E INIT FUNCTION RECEIVED WITHOUT INTERVENING TERM

Explanation: This is an internal error.

User response: Call IBM Software Support.

CKZVE203E UNKNOWN FUNCTION: *xxxx*

Explanation: This is an internal error.

User response: Call IBM Software Support.

CKZVE204E ERROR CALLING CKZ01VV1 *xxxx*
r15=rc r0=rea id=nn

Explanation: This is an internal error.

User response: Call IBM Software Support.

CKZVE205E MLA VALUE FOUND GREATER THAN
4

Explanation: This is an internal error.

User response: Call IBM Software Support.

CKZVE206E MASK FAILED EVALUATION CHECK:
xxxx

Explanation: Mask xxxx is invalid.

User response: Fix the error and retry the function.

**CKZVE207E CSI FAILED WHEN CALLING FOR
MASTER CATALOG NAME**

Explanation: This is an internal error.

User response: Call IBM Software Support.

CKZVE208E MASK FAILED EVALUATION CHECK:
xxxx - yyyy

Explanation: Mask xxxx is invalid for reason yyyy

User response: Fix the error and retry the function.

CKZVE209E UN-ALLOCATION FAILED FOR: cccc

Explanation: Un -allocation failed for catalog cccc.

User response: Determine the reason for and fix the problem, Retry the function.

CKZVE210E ALLOCATION FAILED FOR: cccc

Explanation: Allocation failed for catalog cccc.

User response: Determine the reason for the allocation failure and retry.

**CKZVE211E UCBLLOOK ERROR FOR: VVVV -
RETURN CODE:RC REASON
CODE:RSN**

Explanation: This is an internal error.

User response: Call IBM Software Support.

**CKZVE212E VOLUME NOT MOUNTED: vvvv
DSN=nnn DEVTYPE:ttt**

Explanation: A catalog entry specified that a data set was cataloged to a volume, vvvv, that was not mounted for dsn nnn on device type ttt.

User response: Mount the volume, uncatalog the data set, or respectify the selection masks. Retry the function.

**CKZVE213E IOSCAPU ERROR FOR: VVVV -
RETURN CODE:RC REASON
CODE:RSN**

Explanation: This is an internal error.

User response: Call IBM Software Support.

**CKZVE214E nnnn : VOLSER=vvvv
RC=rrr,REAS=sss,DESC=zzzz**

Explanation: An error occurred when trying to access the VVDS on volume vvvv for data set nnnn . rrr = return code. sss = reason code. zzzz = description when available.

User response: If unable to determine the cause of the error or to fix it, call IBM Software Support for assistance.

**CKZVE215E UCBPIN ERROR FOR: VVVV -
RETURN CODE:RC REASON
CODE:RSN**

Explanation: This is an internal error.

User response: Call IBM Software Support.

**CKZVE216E CVAF ERROR: RETURN CODE:rc
STATUS CODE:sc VOLUME:vvvv
ID=nn PREV FUNC:ffff**

Explanation: This is an internal error.

User response: Call IBM Software Support.

**CKZVE217E ERROR OBTAINING VTOC DATA
FROM vvvv : nnnn**

Explanation: An error occurred when trying to access the VTOC on volume vvvv for data set nnnn.

User response: If unable to determine the cause of the error or to fix it, call IBM Software Support for assistance.

CKZVE218E HSM MCVT NOT FOUND

Explanation: An error occurred when trying to access the HSM control blocks. The most probable cause is HSM has not been started.

User response: If unable to determine the cause of the error or to fix it, call IBM Software Support for assistance.

**CKZVE219E HSM DATA EXTRACT ENDED WITH
ERRORS**

Explanation: An error occurred when trying to access the HSM control blocks. The most probable cause is HSM has not been started.

User response: If unable to determine the cause of the error or to fix it, call IBM Software Support for assistance.

**CKZVE220E MIGRATED DATASET NOT FOUND
IN MCDS: nnnn**

Explanation: The catalog indicated that data set nnnn was migrated, but no entry was found in the HSM MCDS.

User response: If unable to determine the cause of the error or to fix it, call IBM Software Support for assistance.

**CKZVE221W NO CATALOG ENTRIES FOUND FOR
MASK(S)**

Explanation: No catalog entries were found for any of the masks passed to CKZ01VE2.

User response: If this result was not expected, try to determine the reason for the problem and retry the function. If help is needed, call IBM Software Support for assistance.

**CKZVE222E ASSOCIATED ENTRY NOT FOUND IN
CATALOG - nnnn**

Explanation: A non-VSAM alias entry with an association of nnnn. nnnn was not found in the catalog. The most probable cause is an orphaned catalog entry.

User response: Determine the reason for and fix the problem. Retry the function. If help is needed, call IBM Software Support for assistance.

**CKZVE223E UNEXPECTED CATALOG RECORD
CONSTRUCT - ASSOCIATION CELL
NOT WHERE EXPECTED**

Explanation: Catalog entry for entry mentioned is invalid.

User response: Determine the reason for and fix the problem. Retry the function. If help is needed, call IBM Software Support for assistance.

**CKZVE224E TYPE "J" RECORD WITH NO
MATCHING "B" RECORD: nnnn**

Explanation: Catalog entry nnnn is an extension record for a catalog record that does not exist.

User response: Determine the reason for and fix the problem. Retry the function. If help is needed, call IBM Software Support for assistance.

**CKZVE225E ERROR READING SELF DESCRIBING
RECORD FROM CATALOG: nnnn**

Explanation: This is an internal error.

User response: Call IBM Software Support.

**CKZVE226E RECEIVED A "GET" REQUEST
BEFORE RECEIVING A MASK**

Explanation: This is an internal error.

User response: Call IBM Software Support.

**CKZVE227E INPUT CONTROL BLOCK IS NOT
COMPATIBLE**

Explanation: This is an internal error.

User response: Call IBM Software Support.

**CKZVE229E HSM IS NOT ACTIVE - UNABLE TO
RETRIEVE 'REQUESTED DATA**

Explanation: HSM must be active for MCDS data to be retrieved.

User response: Start HSM and retry the function.

**CKZVE230I ALLOCATION FAILED FOR: nnnn -
WILL NOT BE USED FOR SEARCH**

Explanation: Unable to allocate catalog nnnn. No entries that may reside in this catalog will be retrieved.

User response: Informational only. Processing continues.

**CKZVE231E VOLUME NOT MOUNTED: vvvv
DSN=nnn DEVTYPE:ttt**

Explanation: A catalog entry specified that a data set was cataloged to a volume, vvvv, that was not mounted for dsn nnn on device type ttt.

User response: Mount the volume, uncatalog the data set, or respecify the selection masks. Retry the function.

**CKZVE232E NO MCDS DATASET WITH VALID
KEY-RANGE FOUND**

Explanation: An attempt was made to find a dsn in the HSM MCDS but no MCDS found in the HSM address space had a key-range that would accommodate the dsn.

User response: Processing continues. No MCDS data will be included for this data set.

**CKZVE233E NO MCDS DSN(S) FOUND IN HSM
ADDRESS SPACE**

Explanation: An attempt was made to find the data set name(s) of the MCDS(s) allocated to HSM. The attempt failed.

User response: Processing continues. No MCDS data will be included for this data set.

CKZVE234W ERROR CALLING CKZ01HSM
RC=RC

Explanation: This is an internal error.

User response: Processing continues. No MCDS data will be included. If the MEMBER NAME MSCKZVE2 reason for the failure cannot be determined, call IBM Software Support.

CKZVE235E nnnn : VOLSER=vvvvvv - VVDS NOT
OPEN

Explanation: A previous open failed for the VVDS on volume vvvvvv failed.

User response: Processing continues. No VVDS related data is returned for data set nnnn.

CKZVE236W n: Invalid DSN found :
xxxxxxxxxxxxxxxxxxxx

Explanation: The catalog contained an invalid data set name. The first CKZVE236W shows the data set name in character format. The next three CKZVE236W show the name in hex.

User response: Fix the catalog and retry the function.

CKZVE237W OPEN FAILED FOR: nnnn - WILL
NOT BE USED FOR SEARCH

Explanation: Unable to open catalog nnnn. No entries that may reside in this catalog will be retrieved.

User response: Informational only. Processing continues.

CKZVE238E CLUSTER NOT FOUND FOR
TRUENAME IN CATALOG -

Explanation: An error has been detected in your catalog. A VSAM true name record was found without an associated cluster entry.

User response: Informational only. Processing continues. Evaluate catalog entries for the identified data set.

CKZVE239I RETRYING ERROR OBTAINING
?????? FOR dsn - ucat

Explanation: An error occurred when trying to access the VTOC, VVDS, or MCB information for a data set. This message will occur the catalog is incorrect. This message may occur if the data set is being migrated so that the catalog is temporarily out of date.

User response: No action required for this message unless subsequent errors CKZVE217E, CKZCA120E, CKZCA114E, CKZCA220E, or CKZCA214E See action for these messages.

CKZVE251E type error; VOLSER: volser CCHHR:
ccccchhhrr CCHH_LO: cccchhhh
CCHH_HI: cccchhhh POST: xx Loc: lllll

Explanation: An error occurred when trying to access or process the VVDS on the indicated volume. For type of EXCP, an error was returned from a read of a VVDS block. Other types indicate an expected record was not found in the VVDS. lllll is an internal indicator of where the problem occurred.

User response: Contact IBM Software Support. Have available the listing that contains this message.

CKZVSE00E VSAM FAILURE,DDNAME: ddname
VSAM DSN: datasetname or NO DD
ALLOCATION FOUND

Explanation: A VSAM failure occurred attempting to open, access, or close a VSAM data set used by DB2 Cloning Tool.

User response: See additional messages. Although many errors could be caused by internal errors, look for any messages with ** USER CORRECTABLE **. These could include problems such as insufficient region size, unable to allocate extents, etc. Some errors could be the result of incorrectly defining a data set. Compare values returned for catalog LRECL, key length, and key offset to the specifications for the data set as documented in the appropriate product manual. LRECL and KEYLEN to be compared are those supplied with messages CKZVSE21 and CKZVSE22, not CKZVSE19 and CKZVSE20. If the error does not appear to be user correctable, please include all CKZVSEnn messages in documentation supplied to IBM Software Support. Also supply to IBM Software Support the output from a LISTCAT ENT(...) ALL for the failing data set.

CKZVSE01E FAILING CKZ01VSI FUNCTION: nnn
and PREVIOUS CKZ01VSI FUNCTION:
nnn

Explanation: Last and previous (if any) VSAM function performed.

User response: Message provided for product debugging. However, other messages could indicate a user correctable situation. Note that this is the last logical request made by a product program. Because of implicit opens and closes, see message CKZVSE13E to determine the exact VSAM function last requested when determining which set of IBM documented return/reason codes apply.

CKZVSE03E PROGRAM CSECT: csect name

Explanation: Csect name of failing program.

User response: Message provided for debugging. However, other messages could indicate a user correctable situation.

CKZVSE04E VWHEN ERROR ID: id

Explanation: Internally assigned id of last VSAM I/O call.

User response: Message provided for debugging. However, other messages could indicate a user correctable situation.

CKZVSE05E VWHEN ASM LISTING LINE #: line number

Explanation: Source listing line number of last VSAM I/O call.

User response: Message provided for debugging. However, other messages could indicate a user correctable situation.

CKZVSE06E message associated with CKZ01VSI Register 15 value

Explanation: message associated with a non-VSAM error - i.e. a non-zero return code from CKZ01VSI or a VSAM OPEN, ACCESS, or CLOSE.

User response: Message provided for debugging. However, other messages could indicate a user correctable situation. If last function is OPEN, and return code is 16: RLS NOT AVAILABLE - NO SMSVSAM SERVER this may be user correctable if the CKZINI member specified RLS for the data set and in fact RLS is not supported on the image.

CKZVSE07E message associated with VSAM Register 15 value

Explanation: message associated with VSAM Register 15 value.

User response: See message CKZVSE10 and CKZVSE11 for VSAM reason code and description. If CKZVSE12 is produced instead of CKZVSE11, see the IBM manual for a description of the reason code.

CKZVSE08E SVC99 ERROR CODE: code

Explanation: Error code from SVC99 - dynamic allocation.

User response: Use this error code in conjunction with the SVC99 information code (CKZVSE09E) to determine the cause of failure.

CKZVSE09E SVC99 INFORMATION CODE: code

Explanation: Information code from SVC99 - dynamic allocation.

User response: Use this error code in conjunction with the SVC99 error code (CKZVSE08E) to determine the cause of failure.

CKZVSE10E VSAM REASON CODE: code

Explanation: Reason code returned from VSAM open, access, or close.

User response: Use this error code in conjunction with the description provided by message CKZVSE11E to determine the cause of the failure. See additional comments for CKZVSE11E.

CKZVSE11E vsam reason code description

Explanation: Abbreviated description for selected VSAM reason codes.

User response: The error handling module producing CKZVSEnn messages includes many VSAM reason code descriptions. Note that these messages are abbreviated in comparison to the messages as documented by IBM. For this reason, users should also refer to the IBM documentation for the reason code displayed with message CKZVSE10. Because IBM documented reason codes are distinguished by open/close versus access, see message CKZVSE13E for the last access attempted. CKZVSE13E will indicate OP for open, and CL for close. Consider any other value as ACCESS. Note especially any reason code descriptions with the string ** USER CORRECTABLE as these may be situations correctable without IBM Software Support.

CKZVSE12E NO DESCRIPTION FOR REASON CODE

Explanation: Description for reason code not included in error handling table.

User response: Refer to the description of the reason code (CKZVSE10E) for the last function requested (CKZVSE13E) in IBM documentation.

CKZVSE13E VSAM CCODE: code

Explanation: This is a value indicating the last VSAM request issued.

User response: This value is necessary if looking up the return and reason codes in IBM documentation. CCODE will be OP for an open and CL for a close. Consider any other value as ACCESS.

CKZVSE14E CKZ01VSI IMPLICIT OPEN

Explanation: Indicator that last program request required that the data set be re-opened.

User response: Primarily for product debugging.

CKZVSE15E CKZ01VSI IMPLICIT CLOSE

Explanation: Indicator that a close was the result of one task losing control to another.

User response: Primarily for product debugging.

CKZVSE16E CKZ01VSI IMPLICIT REPOSITION

Explanation: Indicator that the positioning had to be re-established as a result of losing control to another task.

User response: Primarily for product debugging.

CKZVSE17E OPEN CLASSIFICATION: value

Explanation: Internal value indicating intended use of the data set.

User response: Primarily for product debugging.

CKZVSE18E BUFFERING TECHNIQUE: value

Explanation: Internal value indicating selected buffering technique.

User response: Primarily for product debugging.

CKZVSE19E CKZ01VSI PARM LRECL: lrecl

Explanation: Last LRECL set in CKZ01VSI parm field.

User response: Primarily for product debugging.

CKZVSE20E CKZ01VSI PARM KEYLEN: key length

Explanation: Last key length set in CKZ01VSI parm field.

User response: Primarily for product debugging.

CKZVSE21E CKZ01VSI CATALOG LRECL: lrecl

Explanation: Max LRECL fetched from catalog at open time.

User response: This should be used to verify that the cluster as defined matches the requirements for the data set as defined in product documentation.

CKZVSE22E CKZ01VSI CATALOG KEYLEN: key length

Explanation: Key length fetched from catalog at open time.

User response: This should be used to verify that the cluster as defined matches the requirements for the data set as defined in product documentation.

CKZVSE23E CKZ01VSI CATALOG KEY OFFSET: key offset

Explanation: Key offset fetched from catalog at open time.

User response: This should be used to verify that the cluster as defined matches the requirements for the data set as defined in product documentation.

CKZVV136E Invalid LRECL: value

Explanation: LRECL must be a positive integer.

User response: correct the LRECL value.

CKZVV137E Dataspace defined over 2 GIG

Explanation: Dataspaces are limited to 2 GIG or 2097152 K. The corresponding SI027_VALUES in CKZINI are specified in K, so the limit is 2097152.

User response: Correct the SI027_Values section of the CKZINI

CKZVV138E Error creating dataspace rc=xx reason yyyyyyyy

Explanation: CKZ01VV1 attempted to create a dataspace. The error indicates that the parameters were incorrect.

User response: Correct the SI027_Values section of the CKZINI

CKZVV139E Error extending dataspace rc=xx reason yyyyyyyy

Explanation: CKZ01VV1 attempted to expand a dataspace. The error indicates that the parameters were incorrect.

User response: Correct the SI027_Values section of the CKZINI

CKZVV140E Bad Token: INITIAL_SIZE

Explanation: Value specified is larger than the value allowed (2097152).

User response: Correct the INITIAL_SIZE value in the CKZINI

CKZVV141E Bad Token: MAXIMUM_SIZE

Explanation: Value specified is larger than the value allowed (2097152).

User response: Correct the MAXIMUM_SIZE value in the CKZINI

CKZVV142E Bad Token:
MAXIMUM_DATASPACE_SIZE

User response: Correct the
MAXIMUM_DATASPACE_SIZE value in the CKZINI.

Explanation: Value specified is larger than the value
allowed (2097152).

How to look up message explanations

You can use several methods to search for messages and codes.

Searching an information center

In the search box that is located in the top left toolbar of any Eclipse help system, such as the IBM Information Management Software for z/OS Solutions Information Center, enter the number of the message that you want to locate. For example, you can enter DFS1065A in the search field.

Use the following tips to improve your message searches:

- You can search for information on codes by entering the code; for example, enter -327.
- Enter the complete or partial message number. You can use the asterisk wildcard character (*) to represent multiple characters, and you can use the question mark wildcard character (?) to represent a single character.

The information center contains the latest message information for all of the Information Management products that are included in the information center.

Searching for messages on the web

You can use any of the popular search engines that are available on the web to search for message explanations. When you type the specific message number or code into the search engine, you are presented with links to the message information in IBM information centers.

Gathering diagnostic information

Before you report a problem with DB2 Cloning Tool to IBM Software Support, you need to gather the appropriate diagnostic information.

Procedure

Provide the following information for all DB2 Cloning Tool problems:

- A clear description of the problem and the steps that are required to re-create the problem
- All messages that were issued as a result of the problem
- Product release number and the number of the last program temporary fix (PTF) that was installed
- The version of DB2 that you are using and the type and version of the operating system that you are using
- For DB2 Cloning Tool Table Space Cloning, if copied objects are inaccessible on the target, provide the create DDL and DSN1PRNT of the page or pages in error.

Provide additional information based on the type of problem that you experienced:

For errors in batch processing, provide the following information:

- The complete job log

- Print output
- Contents of the any data sets that were used during the processing

For online abends, provide the following information:

- A screen shot of the panel that you were using when the abend occurred
- The job log from the TSO session that encountered the abend
- The job log from the server
- A description of the task that you were doing before the abend occurred

Validating load module contents with MODLEVEL

Customer support may request that you submit the MODLEVEL JCL to identify the current modification levels of your product load modules.

MODLEVEL displays internal identifiers that can be used by IBM Software Support to determine the level of the product you are running. This information is based on information stored internally within load modules. Please note that only the last update (level) will contain the fix and revision number and the change date. All other fixes will have only the ID number.

If you experience difficulty running the report, or have any questions, contact IBM Software Support.

JCL for MODLEVEL can be found in member CKZMLVL in the SCKZJCL library. The following is an example of the MODLEVEL JCL.

```
//STEP01 EXEC PGM=CKZ01MOD
//STEPLIB DD DISP=SHR,DSN=CKZ.LOAD (your LOAD library)
//SYSUT1 DD DISP=SHR,DSN=*.STEPLIB
//SYSPRINT DD SYSOUT=*
```

DD statements for the MODLEVEL JCL are as follows:

STEPLIB

Required. Specify the product LOAD library.

SYSPRINT

Required. Output DD statement.

SYSUT1

Required. Specify the primary LOAD library to be processed.

Execution of the MODLEVEL JCL creates a report containing the last fix and a list of fixes for all of the modules in the SYSUT1 library. The following is an example of the DB2 Cloning Tool Subsystem Cloning MODLEVEL report.


```

| CKZMOD01I 09/28/2011 13.38.16 DVLP CKZ01MOD Starting (REV=10,ptf=PRM01060,fixdate=2009/07/24)
| CKZMOD06E 09/28/2011 13.38.16 DVLP CKZ01MOD No Header data available for 'SYSUT1  CKZ$ACES'
| CKZMOD06E 09/28/2011 13.38.16 DVLP CKZ01MOD No Header data available for 'SYSUT1  CKZ$BMSG'
| CKZMOD06E 09/28/2011 13.38.16 DVLP CKZ01MOD No Header data available for 'SYSUT1  CKZ$BOOT'
|
| Highest Build Date is 2011/08/25 13:52
|
| Highest PMR Date is 2011/04/08 16:40  PMR00629
| Highest ETR Date is 2011/08/24 13:34  ETR23082
|
| Member  DD          Last fix  Vers  Rev   Build Date/Time  Macro
|
| CKZ$ACES SYSUT1 ** Non-standard header
|
| CKZ$BMSG SYSUT1 ** Non-standard header
|
| CKZ$BOOT SYSUT1 ** Non-standard header
|
| CKZ00010 SYSUT1 LAST ETR00680  3      32    2011/03/09 17:39 MSCHDR
|           SYSUT1 LIST 16362,16057,16397,16514,16739,16290,17907,17456,
|           SYSUT1 LIST 18311,16289,19527,21924,21276,19773,25006,26790,
|           SYSUT1 LIST 26909,26936,00277,00602,00680
|
| CKZ00011 SYSUT1 LAST ETR00868  V3R1M0 16    2011/03/09 17:40 MSCHDR
|           SYSUT1 LIST 14608,15131,15194,15738,15948,16323,16973,17280,
|           SYSUT1 LIST 17446,17683,19413,21570,19773,23330,26934,00251,
|           SYSUT1 LIST 00868
|
| CKZ00500 SYSUT1 LAST N/A          @DF   0     2011/08/24 13:32 MSCHDR
|
| CKZ00501 SYSUT1 LAST N/A          @02   0     2011/08/24 13:32 MSCHDR
|
| CKZ01VV1 SYSUT1 LAST ETR26666  1.1    27    2011/03/09 10:46 MSCHDR
|           SYSUT1 LIST 09781,10091,10145,11098,11401,11778,12763,13631,
|           SYSUT1 LIST 13631,13873,14582,14584,14585,14740,14864,14877,
|           SYSUT1 LIST 14970,16675,16781,20351,26666
|
| CKZ01VV2 SYSUT1 LAST ETR26636  0101  6     2011/03/09 10:46 MSCHDR
|           SYSUT1 LIST 16384,18817,12226,18917,20082,23548,26636
|
| CKZMOD02I 09/28/2011 13.38.20 DVLP CKZ01MOD Terminating

```

Notices

This information was developed for products and services offered in the U.S.A.

This material may be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
J64A/G4
555 Bailey Avenue
San Jose, CA 95141-1003
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to

IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com)[®] are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at <http://www.ibm.com/legal/copytrade.html>.

Other company, product, and service names may be trademarks or service marks of others.

Privacy policy considerations

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at <http://www.ibm.com/privacy> and the section titled "Cookies, Web Beacons, and Other Technologies" in IBM's Online Privacy Statement at <http://www.ibm.com/privacy/details>. Also, see the "IBM Software Products and Software-as-a-Service Privacy Statement" at <http://www.ibm.com/software/info/product-privacy>.

Index

A

accessibility
 overview 16
ACF authorities 78
ACTIVELOG-LIMIT
 DB2RBLDBSDS 394
APF authorization 28
authorization requirements 20
authorizations 71
AUTO_START_SOURCE_SPACE
 keyword 577
AUTO_START_TARGET_SPACE
 keyword 577
AUTO_STOP_TARGET_SPACE
 keyword 577

B

BACKINFO-DDN
 DB2GETBACKINFO 386
 RESTORE-FROM-DUMPTAPES
 command 451
BACKINFO-REFORMAT command 338
 Command and keyword
 definitions 340
 JCL 343
BACKINFO-REFORMAT command
 syntax 339
BACKUP
 UCATOPTIONS command 457
BACKUP FORCE
 UCATOPTIONS command 457
BCSCLEAN
 target catalog available to
 applications 78
BCSCLEAN command 345
 CLEANUP-CATALOG-
 DSNMASKS 347
 CLEANUP-CATALOG-
 ORPHANS 347
 Command and keyword
 definitions 346
 Delete previous application cycle
 entries from target catalog 345
 JCL 349
 SIMULATE 348
BCSCLEAN command syntax 346
BSDS-DDN
 DB2GETBACKINFO 386

C

Capture catalog data
 COPY step JCL 368
Catalog names
 About source and target 352
catalogs
 populated or empty 78
CATWORK_ATTR keyword 577

CATWORK-ATTR
 COPY command 362
CATWORK-DSN
 COPY command 353
CKZ
 JCLFINDUCATS 427
CKZINI member
 customization values 575
 structure 575
 syntax rules 575
 token errors 577
CKZINI member keywords
 AUTO_START_SOURCE_SPACE 577
 AUTO_START_TARGET_SPACE 577
 AUTO_STOP_TARGET_SPACE 577
 CATWORK_ATTR 577
 CONCURRENT_EXECUTIONS 577
 CONCURRENT_EXECUTIONS_
 WAIT_TIME 577
 COPY_IF_NO_DB2
 _TARGET_OBJECTS 577
 DB2_CLONING_TOOL
 _REL_DATE 577
 DB2_CLONING_TOOL_REL 577
 DB2_PLAN 577
 DB2_XCFCLEAN 577
 DSNS_PER_COPY 577
 DSS_COPY_COMMANDS 577
 GDG_ALL_MIGRATED 577
 GDG_ALL_MIGRATED_RETAIN
 _RC 577
 GDG_EMPTY 577
 GDG_EMPTY_RETAIN_RC 577
 GDG_MIGRATED 577
 GDG_MIGRATED_RETAIN_RC 577
 GDG_TAPE 577
 GDG_TAPE_RETAIN_RC 577
 INIMERGE_VALUES 577
 ISSUE_CKZ14141I 577
 ISSUE_VCLOSE 577
 ISSUE_VCLOSE_SCOPE 577
 MAX_COPY_RC 577
 MAX_RC 577
 MAX_RENAME_TASKS 577
 MAX_SUBTASKS 577
 MIM_GDIF 577
 MISSING_USERCAT_DISP 577
 MISSING_USERCAT_RC 577
 NOT_RENAMED_DISP 577
 NOT_RENAMED_RC 577
 ORPHAN_CATENTRY_DISP 577
 ORPHAN_CATENTRY_RC 577
 RECATALOG 577
 RENAME_ERROR 577
 RENAME_ERROR_CONTINUE
 _RC 577
 REPLACE_TARGET_DSN 577
 RESET_LOGRBA 577
 SPACE_MANAGEMENT 577
 SPECIAL_SECTIONS 577

CKZINI member keywords (*continued*)
 TARGET_VOLS_SHOULD_
 BE_EMPTY 577
 TCPIP_SERVER_PORT 577
 TCPIP_STC_NAME 577
 TEMP_DATASET_DISP 577
 TEMP_DATASET_RC 577
 VTOCIX_REBUILDER 577
CLEANUP-CATALOG-DSNMASKS
 BCSCLEAN command 347
CLEANUP-CATALOG-ORPHANS
 BCSCLEAN command 347
CLIP
 VOLOPTIONS command 470
cloning 95
 DB2 subsystem 75
 subsystem 538, 547, 559, 566, 570
 volumes 535
 with DB2 Administration Tool 335
cloning DB2 subsystem 80
cloning definition 11
cloning from image copies 259
 options for rebuilding indexes 262
 steps 260
cloning from image copies source job
 changes 262
cloning scenarios 535
cluster names, VSAM 75
COLD-START
 DB2ALTERBSDS 375
Command and keyword definitions
 BACKINFO-REFORMAT
 command 340
 BCSCLEAN command 346
 COPY command 353, 476
 COPYCHECK command 372
 DATASUBTYPE command 511
 DB2ALTERBSDS command 375
 DB2FIX command 381
 DB2GETBACKINFO command 386
 DB2LGRNXCLEAN command 391
 DB2RBLDBSDS command 393
 DB2SETLOG command 397
 DB2SQL command 400
 DB2START command 406
 DB2STOP command 409
 DB2UPDATE command 413
 DB2UTILXCLEAN command 421
 DB2XCFCLEAN command 424
 FINDUCATS command 426
 HLQDDDF command 512
 JRNLUPTGRADE command 429
 LISTDEF command 514
 ONLINECLIP command 431
 RENAME command 436
 RESTORE-FROM-DUMPTAPES
 command 451
 SET command 517
 UCATOPTIONS command 456
 VARYOFF command 459
 VARYON command 464

- Command and keyword definitions (continued)
 - VOLOPTIONS command 469
- commands 473
- Commands 337
 - BACKINFO-REFORMAT 338
 - BCSCLEAN 345
 - COPY 351, 473
 - COPYCHECK 371
 - DATASUBTYPE 510
 - DB2ALTERBSDS 374
 - DB2FIX 380
 - DB2GETBACKINFO 385
 - DB2LGRNXCLEAN 390
 - DB2RBLDBSDS 393
 - DB2SETLOG 396
 - DB2SQL 399
 - DB2START 404
 - DB2STOP 409
 - DB2UPDATE 411
 - DB2UTILXCLEAN 420
 - DB2XCFCLEAN 424
 - FINDUCATS 426
 - HLQDDDF 511
 - JRNLUPGRADE 428
 - LISTDEF 512
 - ONLINECLIP 431
 - RENAME 433
 - RESTORE-FROM-DUMPTAPES 450
 - SET 516
 - UCATOPTIONS 456
 - VARYOFF 459
 - VARYON 463
 - VOLOPTIONS 468
- CONCURRENT_EXECUTIONS
 - keyword 577
- CONCURRENT_EXECUTIONS_
 - WAIT_TIME keyword 577
- configuration tasks
 - DB2 version migration
 - considerations 73
- configuring DB2 subsystems for ISPF
 - interface 276, 277
- cookie policy 935, 937
- copies, volumes
 - planning for 75
- COPY 351
 - Initiate volume copies and catalog
 - data capture 368
 - Step JCL 368
- copy by volume 83
- COPY command 473
 - CATWORK-ATTR 362
 - CATWORK-DSN 353
 - Command and keyword
 - definitions 353, 476
 - DATA-MOVER 362, 368
 - DATA-
 - MOVER(PGM(ADRDSSU)) 368
 - EXCLUDE-FROM-VOLSER 366
 - EXCLUDE-TO-VOLSER 366
 - FROM-STORAGEGROUP 357
 - FROM-USER-STORAGEGROUP 357
 - FROM-VOLSER 357
 - FROM-VOLSER-DDN 358
 - JCL 368
 - JOURNAL-DDN 353

- COPY command (continued)
 - JOURNAL-DSN 353
 - Keyword definitions 353
 - NOUSERCATALOGS 356
 - SIMULATE 366
 - Source and target names 352
 - SOURCESONLINE 367
 - Step JCL 368
 - TARGET-VOLS-SHOULD-BE-EMPTY 367
 - TARGETSONLINE 367
 - TARGETSUONLINE 367
 - TO-STORAGEGROUP 358
 - TO-USER-STORAGEGROUP 358
 - TO-VOLSER 359
 - USERCATALOGS 354
 - USERCATALOGS-
 - BACKUPFIRST 356
 - USERCATALOGS-NOBACKUP 355
 - USERSGDEFS-DDN 360
 - USERSGDEFS-OFFSETS 360
 - VOLPAIRS 361
 - VOLPAIRS-DDN 361
 - VOLPAIRSDEVN 361
 - VOLPAIRSDEVN-DDN 361
 - VOLPAIRSDEVN-NOCLIP 361
 - VOLPAIRSDEVN-NOCLIP-DDN 361
- COPY command defaults
 - ISPF interface 280, 299
- COPY command syntax 351, 474
- Copy DB2 Entries panel 66
- COPY step
 - Sample control statements 368
- COPY_IF_NO_DB2_TARGET_OBJECTS
 - keyword 577
- COPY-CMD-DDN
 - VARYOFF command 460
- COPYCHECK 371
 - Use to scan status of volume
 - copies 371
 - Use to withdraw previously
 - established volume
 - relationships 371
- COPYCHECK command
 - Command and keyword
 - definitions 372
 - JCL 373
 - JOURNAL-DDN 372
 - JOURNAL-DSN 372
 - relationships conflicts 78
 - STOPSNAP 373
 - WAIT 372
 - WITHDRAW 372
- COPYCHECK command defaults
 - ISPF interface 280
- COPYCHECK command syntax 371
- coupling facility structures,
 - deallocating 591
- create volume copies 83, 84
- creating a cloning profile
 - ISPF interface 317
- creating cloning jobs
 - ISPF interface 316
- creating XML object table
 - ISPF interface 316
- customization 72

- customization (continued)
 - after customizing using Tools
 - Customizer 71
 - associated list
 - adding DB2 entries 58
 - overview 529
 - associating DB2 entries 58
 - browsing parameters 65
 - changing parameters 51
 - component 529
 - copying DB2 entries 66
 - Create a DB2 Entry panel 58
 - creating DB2 entries 58
 - customization jobs
 - deleting 69
 - displaying 68
 - generating 63
 - maintaining 69
 - regenerating 63
 - renaming 69
 - sort sequence 63
 - submitting 63, 68
 - customization library
 - deleting jobs 69
 - maintaining 69
 - overview 532
 - recustomizing 69
 - renaming jobs 69
 - customization library qualifier
 - specifying 46
 - Customized status 529
 - Customizer Workplace panel 63
 - customizing a new version of a
 - product 51
 - customizing a product for the first
 - time 51
 - customizing settings 46
 - data sets
 - customization library 532
 - data store 532
 - Discover EXEC library 532
 - metadata library 532
 - data store
 - overview 532
 - data store data set
 - specifying 46
 - DB2 data sharing members
 - adding 58
 - associating 58
 - copying 66
 - creating 58
 - DB2 entries 529
 - adding 58
 - associating 58
 - copying 66
 - creating 58
 - defining 63
 - deleting 67, 68
 - generating jobs for 63
 - removing 67
 - selecting 63
 - specifying 63
 - unassociating 67, 68
 - DB2 group attach field
 - specifying 46
 - DB2 group attach names
 - adding 58

- customization (*continued*)
 - DB2 group attach names (*continued*)
 - associating 58
 - copying 66
 - creating 58
 - DB2 parameters
 - defining 61
 - editing 61
 - DB2 Parameters panel 61
 - DB2 subsystems
 - adding 58
 - associating 58
 - copying 66
 - creating 58
 - defining DB2 parameters 61
 - defining parameters 60, 63
 - defining product parameters 60
 - deleting DB2 entries 68
 - deleting jobs 53
 - Discover Customized Product
 - Information panel 56
 - Discover EXEC
 - customizing a new version of a product 51, 52
 - overview 532
 - retrieving product information automatically 56
 - Discovered status 529
 - discovering previous versions 52
 - discovering product information 56
 - displaying jobs 68
 - editing parameters 51
 - editing product parameters 60
 - Errors in Customization status 529
 - Finish Product Customization
 - panel 63
 - first-time 51
 - first-time customization 51
 - generating jobs 63
 - high-level qualifier 529
 - Incomplete status 529
 - job sort order 63
 - jobs
 - deleting 69
 - displaying 68
 - maintaining 69
 - renaming 69
 - sort order 63
 - submitting 63, 68
 - LPARs 69
 - maintaining jobs 69
 - master list
 - adding DB2 entries 58
 - Associate DB2 Entry for Product panel 58
 - overview 529
 - metadata libraries
 - specifying 55
 - metadata library
 - overview 532
 - specifying 46
 - modifying parameters 51
 - modifying settings 46
 - multiple instances 46
 - multiple-LPAR environment 69
 - Not Required status 529

- customization (*continued*)
 - panels
 - Associate DB2 Entry for Product 58
 - Create a DB2 Entry 58
 - Customizer Workplace 63
 - DB2 Parameters 61
 - Discover Customized Product Information 56
 - Finish Product Customization 63
 - Product Parameters 60
 - Specify the Metadata Library 55
 - parameter values 28
 - parameters
 - browsing 65
 - defining 60, 63
 - viewing 65
 - preparing to use Tools
 - Customizer 46
 - product 529
 - product parameters
 - changing 53
 - defining 60
 - editing 53, 60
 - modifying 53
 - Product Parameters panel 60
 - Ready to Customize status 529
 - recustomization 51, 53
 - recustomizing 53
 - recustomizing a product 51
 - removing DB2 entries 67
 - roadmaps 51
 - customizing for the first time 51
 - first-time customization 51
 - reustomizing 53
 - Specify the Metadata Library
 - panel 55
 - specifying data sets 46
 - specifying metadata libraries 55
 - starting Tools Customizer 45
 - status types
 - Customized 529
 - Discovered 529
 - Errors in Customization 529
 - Incomplete 529
 - Not Required 529
 - Ready to Customize 529
 - submitting jobs 63
 - terminology 529
 - user job card settings
 - specifying 46
 - viewing parameters 65
 - customization library
 - overview 532
 - customization library qualifier
 - specifying 46
 - customizing settings 46

D

- data masking
 - rebuilding indexes 241, 250, 251
 - sample job 251
 - specifying masking rules 241
 - summary of steps 239
 - target job changes 250
- data masking source job changes 240

- data set names
 - gathering 25
- data set renaming considerations 78
 - application name length restriction 78
- data set/sphere integrity
 - GDGs 75
 - multivolume data sets 75
 - non-VSAM data set aliases 75
 - VSAM spheres 75
- data sets
 - multivolume 75
 - renamed and not named 75
- data store
 - overview 532
- data store data set
 - specifying 46
- DATA-MOVER
 - COPY command 362, 368
- DATA-MOVER(PGM(ADRDSU))
 - Discrete VOLSER specification 368
 - Source volumes 368
 - Storage group specification 368
 - Target volumes 368
 - VOLSER mask specification 368
- DATABASES
 - DB2FIX 382
- DATABASESONLY
 - DB2GETBACKINFO 386
- DATACLAS
 - RENAME command 437
- DATACLAS-PAIRS
 - RENAME command 437
- DATASUBTYPE 510
- DATASUBTYPE command
 - Command and keyword definitions 511
 - Keyword definitions 511
- DATASUBTYPE command syntax 510
- DB2 10 158
- DB2 Administration Tool, using DB2
 - Cloning Tool with 335
- DB2 Administration Tool for z/OS,
 - adding to Launchpad 72
- DB2 BACKUP SYSTEM 547
- DB2 BACKUP SYSTEM command 538, 559
- DB2 Cloning Tool 337
 - cloning definition 11
 - features and benefits 13
 - ISPF interface 275
 - JCL DB2ALTERBSDS 378, 379
 - JCL DB2SETLOG 398
 - JCL DB2UPDATE 418, 419
 - job templates 269
 - overview 10
 - reasons for using 12
 - support 12
 - why clone a subsystem? 11
- DB2 Cloning Tool Table Space
 - Cloning 473
 - CKZFIX job 178
 - considerations for in-progress read/write activity or DB2 utilities 151
 - copy procedure 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237

DB2 Cloning Tool Table Space Cloning
(*continued*)

- copy procedures 227
- DB2 entity terminology 149
- DB2 Version 9.1 and later
 - considerations 158
- description of 147
- features and benefits 149
- identity columns 152
- index caching 164
- overview 147
- planning for catalog prefetch 163
- planning for cloning 151
- renaming table spaces and index spaces 152
- report job (optional) 176
- rules for using 151
- set up for other methods 199
- set up procedures for FlashCopy, SnapShot, or TimeFinder/Clone 179
- setup for copy by data set 179, 187, 199, 211
- setup for FlashCopy, SnapShot, or TimeFinder/Clone 179, 187
- setup for other copy methods 211
- source job 170
- source job - copy phase IV 172
- source job - discovery phase II 171
- source job - eligibility phase III 171
- source job - init phase I 170
- steps for copying by data set using other methodologies 204, 205, 206, 207, 212, 218, 219, 228
- steps for copying by data set with FlashCopy, SnapShot, or TimeFinder/Clone 179, 180, 188, 189
- steps for copying by data set with FlashCopy/SnapShot 184, 193, 194, 195, 200, 208, 217, 220, 221, 222
- steps for copying by data set with other copy methodologies 211, 222
- steps for copying by data set with other methodologies 199
- subsystem copy rules 151
- target job 174
- TCP/IP server job (optional) 174
- why you should use 148
- DB2 group attach field
 - specifying 46
- DB2 Recovery Expert backup
 - cloning when backup volumes are online 566
- DB2 Recovery Expert for z/OS backup
 - cloning when backup volumes are offline 570
- DB2 subsystem cloning 538, 559, 566, 570
 - cloning to a specific point in time using a BACKUP SYSTEM backup 547
- DB2 Version 9.1 158
- DB2 version migration considerations 73
- DB2_CLONING_TOOL keywords 577
- DB2_CLONING_TOOL_REL keywords 577
- DB2_PLAN keyword 577
- DB2_XCFCLEAN keyword 577
- DB2-MEMBER
 - DB2ALTERBSDS 375
- DB2-NAME
 - DB2LGRNXCLEAN 391
- DB2-SSID
 - DB2FIX 382
- DB2ALTERBSDS 374
- DB2ALTERBSDS command
 - COLD-START 375
 - Command and keyword definitions 375
 - DB2-MEMBER 375
 - JCL 377
 - JOURNAL-DDN 375
 - JOURNAL-DSN 375
 - Step JCL 378, 379
- DB2ALTERBSDS command syntax 374
- DB2FIX command
 - Command and keyword definitions 381
 - DATABASES 382
 - DB2-SSID 382
 - DSNDB01-DBD01-STARTED 382
 - EXCLUDE-MASKS 382
 - JCL 384
 - MAX-CONCURRENT-CMDS 382
 - MEMBERS-AND-DBD01 382
 - MEMBERS-NEED-STARTING 383
 - SIMULATE 383
 - START-SCOPE 383
 - WAIT 383
 - WAIT-AND-DBD01 384
- DB2FIX command defaults
 - ISPF interface 285
- DB2FIX command syntax 381
- DB2GETBACKINFO command
 - BACKINFO-DDN 386
 - BSDS-DDN 386
 - Command and keyword definitions 386
 - DATABASESONLY 386
 - DUMP-CLASS 387
 - JCL 388
 - LAST 387
 - LIST 387
 - LOCATION 386
 - LOGONLY 387
 - TOKEN 387
 - USE-DUMPTAPES 387
 - USERCATALOGS 388
 - WORK-DDN 386
 - WORK-DSN 386
- DB2GETBACKINFO command syntax 385
- DB2LGRNXCLEAN
 - JOURNAL-DSN 391
- DB2LGRNXCLEAN command
 - Command and keyword definitions 391
 - DB2-NAME 391
 - JCL 392
- DB2LGRNXCLEAN command syntax 391
- DB2RBLDBSDS
 - JOURNAL-DSN 394
- DB2RBLDBSDS command
 - ACTIVELOG-LIMIT 394
 - Command and keyword definitions 393
 - JCL 395
- DB2RBLDBSDS command syntax 393
- DB2SETLOG command
 - Command and keyword definitions 397
 - JCL 397
 - Step JCL 398
- DB2SETLOG command syntax 396
- DB2SQL command
 - Command and keyword definitions 400
 - JCL 403
- DB2SQL command defaults
 - ISPF interface 286
- DB2START command
 - Command and keyword definitions 406
 - JCL 408
- DB2START command defaults
 - ISPF interface 286
- DB2START command syntax 405
- DB2STOP command
 - Command and keyword definitions 409
 - JCL 411
- DB2STOP command defaults
 - ISPF interface 286
- DB2STOP command syntax 409
- DB2UPDATE command
 - Command and keyword definitions 413
 - JCL 417
 - Step JCL 418, 419
- DB2UPDATE command defaults
 - ISPF interface 287
- DB2UPDATE command syntax 412
- DB2UTILXCLEAN command
 - Command and keyword definitions 421
 - JCL 422
 - JCL for identifying objects with UT status 423
- DB2UTILXCLEAN command defaults
 - ISPF interface 287
- DB2UTILXCLEAN command syntax 421
- DB2XCFCLEAN command
 - Command and keyword definitions 424
 - JCL 425
- DB2XCFCLEAN command syntax 424
- DBSQL command syntax 399
- DDL, processing 160
- DDL, scenarios for processing 161
- deallocating coupling facility
 - structures 591
- dedicated volumes
 - conserve space 75
- DEFAULT_IF_NO_MATCH
 - RENAME command 437, 447
- Delete previous application cycle entries using BCSCLEAN 345
- diagnostic information
 - gathering 931

DIRECTION
 HLQDDDF command 512
 Discover EXEC
 overview 532
 discrete vols vs.SMS storage group
 specs 75
 Discrete VOLSER specification
 DATA-
 MOVER(PGM(ADRDSSU)) 368
 documentation
 accessing 14
 sending feedback 14
 documentation changes 1
 DRIVEACS
 RENAME command 437
 DSNDB01-DBD01-STARTED
 DB2FIX 382
 DSNS_PER_COPY keyword 577
 DSS_COPY_COMMANDS keyword 577
 dump tapes, using for cloning 559
 DUMP-CLASS
 DB2GETBACKINFO 387

E

EBCDIC code set 22, 150
 EMC TimeFinder/Clone Mainframe Snap
 Facility data set level support 147
 EMC TimeFinder/Mirror target volume
 clones 90
 EMC TimeFinder/Mirror Target Volume
 Clones 90
 EMCAPI
 cloning table spaces with 509
 GLOBAL command parameter values
 for 509
 empty ICF catalog 78
 EXCLUDE-FROM-VOLSER
 COPY command 366
 FINDUCATS command 427
 EXCLUDE-MASKS
 DB2FIX 382
 EXCLUDE-SRCNAME
 RENAME command 438
 EXCLUDE-SRCNAME-MASKS
 RENAME command 438
 EXCLUDE-SRCNAME-MASKS-DDN
 RENAME command 438
 EXCLUDE-TO-VOLSER
 COPY command 366
 RESTORE-FROM-DUMPTAPES
 command 451
 EXCLUDEFROMVOLSER
 FINDUCATS command 427

F

Filter masks
 Newname syntax 435
 Oldname syntax 434
 Oldvalue syntax 340
 Rename considerations 434
 filter pattern masks 23
 FINDUCATS 426

FINDUCATS command
 Command and keyword
 definitions 426
 EXCLUDE-FROM-VOLSER 427
 EXCLUDEFROMVOLSER 427
 FROM-STORAGEGROUP 426
 FROM-VOLSER 427
 FROMSTORAGEGROUP 426
 FROMVOLSER 427
 JCL 427
 Keyword definitions 426
 Step JCL 427
 FINDUCATS command syntax 426
 first-time customization 51
 Flash Copy
 source and target volume pairing 75
 FlashCopy 83, 147
 FROM-STORAGEGROUP
 COPY command 357
 FINDUCATS command 426
 FROM-USER-STORAGEGROUP
 COPY command 357
 FROM-VOLSER
 COPY command 357
 FINDUCATS command 427
 FROM-VOLSER-DDN
 COPY command 358
 FROMSTORAGEGROUP
 FINDUCATS command 426
 FROMVOLSER
 FINDUCATS command 427
 function authorization requirements 20

G

GDG_ALL_MIGRATED keyword 577
 GDG_ALL_MIGRATED_RETAIN_RC
 keyword 577
 GDG_EMPTY keyword 577
 GDG_EMPTY_RETAIN_RC
 keyword 577
 GDG_MIGRATED keyword 577
 GDG_MIGRATED_RETAIN_RC
 keyword 577
 GDG_TAPE keyword 577
 GDG_TAPE_RETAIN_RC keyword 577
 GDG-ALL-MIGRATED
 RENAME command 439
 GDG-EMPTY
 RENAME command 439
 GDG-MIGRATED
 RENAME command 439
 GDG-TAPE
 RENAME command 440
 GDGs
 base names matching rename
 mask 75
 source volume list 75

H

HLQDDDF 511
 HLQDDDF command
 Command and keyword
 definitions 512
 DIRECTION 512

HLQDDDF command (*continued*)
 Keyword definitions 512
 HLQDDDF command defaults
 ISPF interface 315
 HLQDDDF command syntax 511

I

image copies, using as cloning
 source 259, 260, 262
 INI keywords
 SI027_VALUES 577
 SI040_VALUES 577
 INIMERGE_VALUES keyword 577
 ISPF interface 275
 about 275
 configuring DB2 subsystems 276, 277
 configuring user settings 278
 HLQDDDF DD data definitions 316
 setting subsystem cloning
 defaults 279
 setting user defaults 278
 specifying subsystem cloning
 information 277
 specifying table space cloning
 information 278
 table space cloning DD
 descriptions 291
 table space cloning defaults
 ADVISORY-STATUS-VALUES 297
 CATALOG-PREFETCH 307
 DDL-ATTRIBUTE-CHANGE
 defaults 309
 DDs 289
 job template data set and member
 names 309
 job template variables 298
 LOG-APPLY defaults 311
 OBJECT-MISMATCH-RETURN-
 CODE 315
 OBJECT-TRANSLATE 308
 RESTRICT-STATUS-VALUES 297
 ISSUE_CKZ14141I keyword 577
 ISSUE_VCLOSE keyword 577
 ISSUE_VCLOSE_SCOPE keyword 577
 ISSUE-VCLOSE
 RENAME command 440

J

JCL
 BACKINFO-REFORMAT 343
 BCSCLEAN 349
 COPY 368
 COPYCHECK 373
 DB2ALTERBSDS 377
 DB2ALTERBSDS step
 Sample control statements 378,
 379
 DB2FIX 384
 DB2GETBACKINFO 388
 DB2LGRNXCLEAN 392
 DB2RBLDBSDS 395
 DB2SETLOG 397
 DB2SETLOG step
 Sample control statements 398

JCL (continued)

- DB2SQL 403
- DB2START 408
- DB2STOP 411
- DB2UPDATE 417
- DB2UPDATE step
 - Sample control statements 418, 419
- DB2UTILXCLEAN 422
- DB2UTILXCLEAN identify objects
 - with UT status 423
- DB2XCFCLEAN 425
- FINDUCATS 427
- JRNLUPLGRADE 430
- ONLINECLIP 432
- RENAME 448
- RESTORE-FROM-DUMPTAPES 454
- UCATOPTIONS 458
- VARYOFF 461
- VARYON 466
- VOLOPTIONS 471
- job template variables 269, 271, 272
- job templates 269
- job templates for utilities 272
- Journal data set
 - Upgrade 428
- JOURNAL-DDSN
 - COPY 353
 - COPYCHECK 372
 - DB2ALTERBDS 375
 - JRNLUPLGRADE 429
 - ONLINECLIP 431
 - RENAME 436
 - VARYOFF 460
 - VARYON 464
 - VOLOPTIONS 469
- JOURNAL-DSN
 - COPY 353
 - COPYCHECK 372
 - DB2ALTERBDS 375
 - DB2LGRNXCLEAN 391
 - DB2RBLDBSDS 394
 - JRNLUPLGRADE 429
 - ONLINECLIP 431
 - RENAME 436
 - VARYOFF 460
 - VARYON 464
 - VOLOPTIONS 469
- JRNLUPLGRADE 428
 - Command scenarios 428
- JRNLUPLGRADE command
 - Command and keyword definitions 429
- JCL 430
- JOURNAL-DDN 429
- JOURNAL-DSN 429
 - Keyword definitions 429
 - OLD-JOURNAL-DDN 429
 - OLD-JOURNAL-DSN 429
- JRNLUPLGRADE command syntax 429

K

- Keyword definitions
 - BCSCLEAN command 346
 - COPY
 - JOURNAL-DSN 353

Keyword definitions (continued)

- COPY command 353
- COPYCHECK
 - JOURNAL-DSN 372
- DATASUBTYPE command 511
- DB2ALTERBDS
 - JOURNAL-DSN 375
- FINDUCATS command 426
- HLQDDDF command 512
- JRNLUPLGRADE
 - JOURNAL-DSN 429
 - OLD-JOURNAL-DSN 429
- JRNLUPLGRADE command 429
- ONLINECLIP
 - JOURNAL-DSN 431
- ONLINECLIP command 431
- RENAME
 - JOURNAL-DSN 436
- RENAME command 436
- VARYOFF
 - JOURNAL-DSN 460
- VARYOFF command 459
- VARYON
 - JOURNAL-DSN 464
- VARYON command 464
- VOLOPTIONS
 - JOURNAL-DSN 469
- VOLOPTIONS command 469

L

- LAST 387
- Launchpad, adding DB2 Cloning Tool 72
- legal notices
 - cookie policy 935, 937
 - notices 935
 - programming interface information 935
 - trademarks 935, 937
- LIST
 - DB2GETBACKINFO 387
 - UCATOPTIONS command 457
 - VOLOPTIONS command 470
- LISTDEF 512
- LISTDEF command
 - Command and keyword definitions 514
- LISTDEF command syntax 513
- load libraries
 - APF authorizing 28
- LOCATION
 - DB2GETBACKINFO 386
- LOG-APPLY options for rebuilding indexes 267
- LOG-APPLY source job changes 266
- LOG-APPLY source TCP/IP server job changes 266
- LOG-APPLY target job changes 267
- LOGONLY
 - DB2GETBACKINFO 387

M

- maintenance 75
- masks 23

- MAX_COPY_RC keyword 577
- MAX_RC keyword 577
- MAX_RENAME_TASKS keyword 577
- MAX_SUBTASKS keyword 577
- MAX-CONCURRENT-CMDS
 - DB2FIX 382
- MAX-TAPEDRIVES
 - RESTORE-FROM-DUMPTAPES command 451
- MAX-TASKS
 - RENAME command 441
- MAX-VOLS-PER-CMD
 - VARYOFF command 460
- MAX-VOLS-PER-CMD command
 - VARY-DDN 465
- MEMBERS-AND-DBD01
 - DB2FIX 382
- MEMBERS-NEED-STARTING
 - DB2FIX 383
- message conventions 592
- messages
 - methods for accessing 931
- metadata library
 - overview 532
 - specifying 46
- MGMTCLAS
 - RENAME command 441
- MGMTCLAS-PAIRS
 - RENAME command 441
- migrated application data set 75
- MIM_GDIF keyword 577
- mirror image 12
- MISSING_USERCAT_DISP keyword 577
- MISSING_USERCAT_RC keyword 577
- MISSINGUCAT
 - RENAME command 442
- modifying settings 46
- MODLEVEL 932
- multivolume data sets 75

N

- NEWCATWORKS
 - UCATOPTIONS command 457
- NEWCATWORKS-DDN
 - UCATOPTIONS command 457
- Newname masks
 - Syntax 435
- NEWTARGETS
 - UCATOPTIONS command 458
 - VOLOPTIONS command 471
- NEWTARGETS-DDN
 - UCATOPTIONS command 458
 - VOLOPTIONS command 471
- NEWTARGETSDEVN
 - VOLOPTIONS command 471
- non-VSAM data set aliases
 - rename masks 75
- NOT_RENAMED_DISP keyword 577
- NOT_RENAMED_RC keyword 577
- notices 935
- NOTRENAMED
 - RENAME command 442
- NOUSERCATALOGS
 - COPY command 356

O

OBJECT-TRANSLATE
 considerations for 504
offline cloning 95
offline cloning steps 96, 99, 102
offline cloning steps summary 96, 98, 101
Offline volume
 With DATA-
 MOVER(PGM(ADRDSU)) 368
OFFLINECLIP
 VOOPTIONS command 470
OLD-JOURNAL-DDN
 JRNLUPTGRADE 429
OLD-JOURNAL-DSN
 JRNLUPTGRADE 429
Oldname masks
 Syntax 434
Oldvalue masks
 Syntax 340
online cloning 104
online cloning steps 107
online cloning steps summary 105
online cloning steps summary, remove data sharing members 111
online cloning steps summary, target becoming non-data sharing 117
online cloning steps, removal of data sharing members 113
online cloning steps, target becoming non-data sharing 118
online status of target volumes 75
ONLINECLIP 431
ONLINECLIP command
 Command and keyword definitions 431
 JCL 432
 JOURNAL-DDN 431
 JOURNAL-DSN 431
 Keyword definitions 431
 WRONG-VOLSER 432
ONLINECLIP command syntax 431
onsite mirror 87
ORPHAN_CATENTRY_DISP
 keyword 577
ORPHAN_CATENTRY_RC
 keyword 577
ORPHANCATENTRY
 RENAME command 443
overview 1

P

pairing source and target volumes 75
panels
 Copy DB2 Entries 66
parameters
 customization 28
pattern masks, filtering 23
planning vol copies and renames
 discrete vols vs. SMS storage group 75
populated ICF catalog 78
preparing to use Tools Customizer 46
problems
 diagnostic information about 931

procedures
 cloning DB2 subsystem 80
PROCESS-DDLDDL-ATTRIBUTE-CHANGE
 DDL-ATTRIBUTE-CHANGE 506
processing DDL 160
product authorizations 71
PRODUCT_INFO keywords 577
programming interface information 935

R

RACF authorities 78
reader comment form 14
RECATALOG
 RENAME command 443
 target catalog renaming considerations 78
RECATALOG keyword 577
remove data sharing members 98
RENAME 433
RENAME command
 Command and keyword definitions 436
 DATACLAS 437
 DATACLAS-PAIRS 437
 DEFAULT_IF_NO_MATCH 437, 447
 DRIVEACS 437
 EXCLUDE-SRCNAME 438
 EXCLUDE-SRCNAME-MASKS 438
 EXCLUDE-SRCNAME-MASKS-DDN 438
 GDG-ALL-MIGRATED 439
 GDG-EMPTY 439
 GDG-MIGRATED 439
 GDG-TAPE 440
 ISSUE-VCLOSE 440
 JCL 448
 JOURNAL-DDN 436
 JOURNAL-DSN 436
 Keyword definitions 436
 MAX-TASKS 441
 MGMTCLAS 441
 MGMTCLAS-PAIRS 441
 MISSINGUCAT 442
 NOTRENAMED 442
 ORPHANCATENTRY 443
 RECATALOG 443
 RENAME-AUDIT-LOG 444
 RENAME-ERROR 444
 RENAME-LIST 445
 RENAME-MASKS 436
 RENAME-MASKS-DDN 437
 RERUN 445
 SAFE 445
 SIMULATE 446
 SOURCE 447
 SPEED 446
 STORCLAS 446
 STORCLAS-PAIRS 447
 TEMPDSN 447
 VOLBKUP-DDN 448
RENAME command defaults
 ISPF interface 281
RENAME command syntax 433
Rename considerations
 Filter masks 434
Rename considerations (*continued*)
 Rename masks 434
rename masks 75, 78
 GDG base names 75
 non-VSAM data set aliases 75
 VSAM cluster names 75
Rename masks
 Newname syntax 435
 Oldname syntax 434
 Oldvalue syntax 340
 Rename considerations 434
rename masks and existing naming conventions 78
RENAME_ERROR keyword 577
RENAME_ERROR_CONTINUE_RC keyword 577
RENAME-AUDIT-LOG
 RENAME command 444
RENAME-ERROR
 RENAME command 444
RENAME-LIST
 RENAME command 445
RENAME-MASKS
 RENAME command 436
RENAME-MASKS-DDN
 RENAME command 437
REPLACE_TARGET_DSN keyword 577
requirements
 source and target volume pairing 75
RERUN
 RENAME command 445
 RESTORE-FROM-DUMPTAPES command 452
RESET_LOGRBA keyword 577
RESTORE-FROM-DUMPTAPES 450
RESTORE-FROM-DUMPTAPES command
 BACKINFO-DDN 451
 Command and keyword definitions 451
 EXCLUDE-TO-VOLSER 451
 JCL 454
 MAX-TAPEDRIVES 451
 RERUN 452
 SIMULATE 452
 SOURCE-STORAGEGROUP 452
 STATUS-DDN 452
 TAPE-UNIT 453
 TO-STORAGEGROUP 451
 TO-VOLSER 451
 USERCATALOGS 453
 USERCATALOGS-DDN 453
 VARY-SCOPE 453
 VOLPAIRS-DDN 453
RESTORE-FROM-DUMPTAPES command defaults
 ISPF interface 288
RESTORE-FROM-DUMPTAPES command syntax 450
RESUME
 VOOPTIONS command 470
return code
 changing value 80
roadmaps
 customizing for the first time 51
 first-time customization 51
RVA DASD 12

RVA DASD (*continued*)
 RVA partition 75
 source and target volume pairing 75
 track format 75

S

SAFE
 RENAME command 445
 SCOPE
 VARYOFF command 460
 SCOPE command
 VARY-DDN 465
 screen readers and magnifiers 16
 security 71
 selection of source/target volumes
 discrete vols vs. SMS storage
 group 75
 service information 14
 SET 516
 SET command
 Command and keyword
 definitions 517
 SET command defaults
 ISPF interface 293
 SET command syntax 516
 setting table space cloning defaults
 ISPF interface 288
 SI027_VALUES 577
 SI040_VALUES 577
 SIMULATE
 BCSCLEAN command 348
 COPY command 366
 DB2FIX 383
 RENAME command 446
 RESTORE-FROM-DUMPTAPES
 command 452
 VARYOFF command 461
 VOLOPTIONS command 470
 SIMULATE command
 VARY-DDN 465
 SMS class values 400
 SnapShot 83, 147
 source and target volume pairing 75
 SOURCE
 RENAME command 447
 VARYOFF command 460
 VARYON command 464
 source and target volume
 migrated application data set 75
 source and target volume pairing 75
 source and target volumes
 pairing and requirements 75
 source volume
 condition 75
 data set/sphere integrity 75
 Source volume
 Names 352
 Source volumes
 Offline volumes 368
 SOURCE-STORAGEGROUP
 RESTORE-FROM-DUMPTAPES
 command 452
 SOURCESONLINE
 COPY command 367
 SPACE_MANAGEMENT 577
 SPECIAL_SECTIONS keyword 577

specifying data sets 46
 SPEED
 RENAME command 446
 START-SCOPE
 DB2FIX 383
 status of inflight transactions 104
 STATUS-DDN
 RESTORE-FROM-DUMPTAPES
 command 452
 STOPSNAP
 COPYCHECK 373
 storage group specification 75
 Storage group specification
 DATA-
 MOVER(PGM(ADRDSU)) 368
 storage management 75
 STORCLAS
 RENAME command 446
 STORCLAS-PAIRS
 RENAME command 447
 stored procedure
 about 123
 example of cloning a non data sharing
 DB2 system 139
 example of cloning a non data sharing
 DB2 system from a DB2 system-level
 backup 140
 example of using the stored procedure
 to clone a non data sharing
 subsystem from DB2 system-level
 backup dump tapes 142
 example of using the stored procedure
 to clone a non data sharing
 subsystem from DB2 system-level
 backup dump tapes across
 sysplexes 143
 JCL members 137
 parameter files and parameter
 descriptions 124
 recloning a DB2 system 138
 running the verified cloning 138
 steps for 124
 syntax 135
 verifying the cloning 137
 stored procedure, using 123
 subsystem cloning 535, 538, 566, 570
 building the cloning jobs 322
 cloning job reference 323
 ISPF interface 319, 322, 323, 327
 planning for 75
 selecting source and target
 subsystems 319
 specify rename masks 322
 specify source and target ICF
 catalogs 322
 submitting the cloning jobs 327
 subsystem cloning step summary
 ISPF interface 318
 subsystem cloning steps
 ISPF interface 317
 subsystem cloning to a specific point in
 time from a BACKUP SYSTEM
 backup 547
 subsystem cloning using SLB dump
 tapes 559
 subsystem, cloning 80
 summary of changes 1

support
 required information 931
 support information 14
 supported volume copy products
 Flash Copy 75
 FlashCopy 12
 SnapShot 12, 75
 syntax
 keywords in CKZINI member 577
 Syntax
 BACKINFO-REFORMAT
 command 339
 BCSCLEAN command 346
 COPY command 351, 474
 COPYCHECK command 371
 DATASUBTYPE command 510
 DB2ALTERBSDS command 374
 DB2FIX command 381
 DB2GETBACKINFO command 385
 DB2LGRNXCLEAN command 391
 DB2RBLDBSDS command 393
 DB2SETLOG command 396
 DB2START command 405
 DB2STOP command 409
 DB2UPDATE command 412
 DB2UTILXCLEAN command 421
 DB2XCFCLEAN command 424
 DBSQL command 399
 FINDUCATS command 426
 HLQDDDF command 511
 JRNLUPGRADE command 429
 LISTDEF command 513
 Newname masks 435
 Oldname masks 434
 Oldvalue masks 340
 ONLINECLIP command 431
 RENAME command 433
 considerations for 434
 RESTORE-FROM-DUMPTAPES
 command 450
 SET command 516
 UCATOPTIONS command 456
 VARYOFF command 459
 VARYON command 464
 VOLOPTIONS command 469
 syntax rules, CKZINI member 575

T

table space cloning
 APF authorization 154
 building the table space cloning
 jobs 332
 caching indexes using catalog
 prefetch 164
 considerations for copying versioned
 objects 166
 considerations for FUZZY-copy 160
 considerations for in-progress
 read/write activity or DB2
 utilities 151
 considerations for renaming data
 sets 159
 considerations for target objects
 created using DEFINE NO 155
 copy procedure for all other copy
 methodologies 231, 234

- table space cloning (*continued*)
 - copy procedure for FlashCopy, SnapShot, or TimeFinder/Clone 227, 228
 - copy products used 147
 - DDF and TCP/IP considerations 153
 - dropped and altered tables 166
 - editing the source job 329
 - fast replication tools 147
 - function overview 169
 - high level qualifiers 159
 - how table spaces and index spaces are stopped and started 157, 158
 - identity columns 152
 - ISPF interface 328, 329, 330, 331, 332, 333
 - LOBs, BLOBs, CLOBs, and DBCLOBs 152
 - missing target DB2 objects 155
 - non-SMS managed volumes 156
 - object attributes 154
 - overview 147
 - planning for catalog prefetch 163
 - planning for cloning 151
 - plans and packages 166
 - RACF and ACF authorities 154
 - renaming table spaces and index spaces 152
 - return code choices 166
 - rules for using 151
 - selecting source and target subsystems 328
 - source job overview 169
 - space processing order 156
 - submitting the cloning jobs 333
 - subsystem copy rules 151
 - target data set ICF catalog aliases 159
 - target job overview 169
 - TCP/IP job overview 170
 - verify the report job settings 330
 - verify the TCP/IP server job settings 331
 - verifying the target job settings 330
 - XML 152
- table space cloning step summary
 - ISPF interface 328
- table space cloning steps
 - ISPF interface 327
- TAPE-UNIT
 - RESTORE-FROM-DUMPTAPES command 453
- TARGET
 - VARYOFF command 460
 - VARYON command 464
- target becoming non-data sharing 101, 102
- Target catalog
 - Delete previous application cycle entries using BCSCLEAN 345
- target catalog renaming considerations
 - RECATALOG option 78
- target volume
 - condition 75
 - data set/sphere integrity 75
- Target volume names 352
- target volumes
 - online status 75
- Target volumes
 - Offline volumes 368
- TARGET_VOLS_SHOULD_BE_EMPTY keywords 577
- TARGET-VOLS-SHOULD-BE-EMPTY
 - COPY command 367
- TARGETSONLINE
 - COPY command 367
- TARGETSUONLINE
 - COPY command 367
- TCPIP_SERVER_PORT keyword 577
- TCPIP_STC_NAME keyword 577
- technotes 14
- TEMP_DATASET_DISP keyword 577
- TEMP_DATASET_RC keyword 577
- TEMPDSN
 - RENAME command 447
- terminology for "objects" 149
- TO-STORAGEGROUP
 - COPY command 358
 - RESTORE-FROM-DUMPTAPES command 451
- TO-USER-STORAGEGROUP
 - COPY command 358
- TO-VOLSER
 - COPY command 359
 - RESTORE-FROM-DUMPTAPES command 451
- TOKEN
 - DB2GETBACKINFO 387
- Tools Customizer
 - associated list
 - adding DB2 entries 58
 - overview 529
 - associating DB2 entries 58
 - browsing parameters 65
 - component 529
 - Copy DB2 Entries panel 66
 - copying DB2 entries 66
 - Create a DB2 Entry panel 58
 - creating DB2 entries 58
 - customization jobs
 - deleting 69
 - displaying 68
 - generating 63
 - maintaining 69
 - renaming 69
 - sort sequence 63
 - submitting 63, 68
 - customization library
 - deleting jobs 69
 - maintaining 69
 - recustomizing 69
 - renaming jobs 69
 - customization library qualifier
 - specifying 46
 - Customized status 529
 - Customizer Workplace panel 63
 - customizing a new version of a product 51, 52
 - customizing a product for the first time 51
 - data sets
 - customization library 532
 - data store 532
- Tools Customizer (*continued*)
 - data sets (*continued*)
 - Discover EXEC library 532
 - metadata library 532
 - data store data set
 - specifying 46
 - DB2 data sharing members
 - adding 58
 - associating 58
 - copying 66
 - creating 58
 - DB2 entries 529
 - adding 58
 - associating 58
 - copying 66
 - creating 58
 - defining 63
 - deleting 67, 68
 - generating jobs for 63
 - removing 67
 - selecting 63
 - specifying 63
 - unassociating 67, 68
 - DB2 group attach field
 - specifying 46
 - DB2 group attach names
 - adding 58
 - associating 58
 - copying 66
 - creating 58
 - DB2 parameters
 - defining 61
 - editing 61
 - DB2 Parameters panel 61
 - DB2 subsystems
 - adding 58
 - associating 58
 - copying 66
 - creating 58
 - defining DB2 parameters 61
 - defining parameters 60, 63
 - defining product parameters 60
 - deleting DB2 entries 68
 - deleting jobs 53
 - Discover Customized Product
 - Information panel 56
 - Discover EXEC
 - customizing a new version of a product 51, 52
 - retrieving product information automatically 56
 - Discovered status 529
 - discovering product information 56
 - displaying jobs 68
 - editing product parameters 60
 - Errors in Customization status 529
 - Finish Product Customization
 - panel 63
 - first-time customization 51
 - generating jobs 63
 - high-level qualifier 529
 - Incomplete status 529
 - job sort order 63
 - jobs
 - deleting 69
 - displaying 68
 - maintaining 69

Tools Customizer (*continued*)

- jobs (*continued*)
 - renaming 69
 - submitting 68
- maintaining jobs 69
- master list
 - adding DB2 entries 58
 - Associate DB2 Entry for Product panel 58
 - overview 529
- metadata libraries 55
- specifying 55
- metadata library
 - specifying 46
- multiple instances 46
- multiple-LPAR environment 69
- Not Required status 529
- panels
 - Associate DB2 Entry for Product 58
 - Copy DB2 Entries 66
 - Create a DB2 Entry 58
 - Customizer Workplace 63
 - DB2 Parameters 61
 - Discover Customized Product Information 56
 - Finish Product Customization 63
 - Product Parameters 60
 - Specify the Metadata Library 55
- parameters
 - browsing 65
 - viewing 65
- preparing to use 46
- product 529
- product parameters
 - changing 53
 - editing 53
 - modifying 53
- Product Parameters panel 60
- Ready to Customize status 529
- recustomization 51
- recustomizing a product 51, 53
- removing DB2 entries 67
- roadmaps
 - customizing a new version of a product 52
 - recustomizing a product 53
 - using the Discover EXEC 52
- Specify the Metadata Library panel 55
- specifying metadata libraries 55
- starting 45
- status types
 - Customized 529
 - Discovered 529
 - Errors in Customization 529
 - Incomplete 529
 - Not Required 529
 - Ready to Customize 529
- submitting jobs 63
- terminology 529
- user job card settings
 - specifying 46
 - using the Discover EXEC 52
 - viewing parameters 65
- trademarks 935, 937
- transactions in flight 104

U

- U.S.A. EBCDIC code set 22, 150
- UCATOPTIONS 456
- UCATOPTIONS command
 - BACKUP 457
 - BACKUP FORCE 457
 - Command and keyword definitions 456
 - JCL 458
 - LIST 457
 - NEWCATWORKS 457
 - NEWCATWORKS-DDN 457
 - NEWTARGETS 458
 - NEWTARGETS-DDN 458
 - UPDATE 457
- UCATOPTIONS command syntax 456
- UPDATE
 - UCATOPTIONS command 457
- USE-DUMPTAPES
 - DB2GETBACKINFO 387
- user job card settings
 - specifying 46
- user-created clones 87
- user-created target volumes 88
- USERCATALOGS
 - COPY command 354
 - DB2GETBACKINFO 388
 - RESTORE-FROM-DUMPTAPES command 453
- USERCATALOGS-BACKUPFIRST
 - COPY command 356
- USERCATALOGS-DDN
 - RESTORE-FROM-DUMPTAPES command 453
- USERCATALOGS-NOBACKUP
 - COPY command 355
- USERSGDEFS-DDN
 - COPY command 360
- USERSGDEFS-OFFSETS
 - COPY command 360

V

- validating load module contents 932
- VARY-DDN
 - MAX-VOLS-PER-CMD command 465
 - SCOPE command 465
 - SIMULATE command 465
 - VARYOFF command 460
 - VARYON command 465
 - VOL-ALREADY-ONLINE command 465
 - WRONG-VOLSER command 466
- VARY-SCOPE
 - RESTORE-FROM-DUMPTAPES command 453
- VARYOFF 459
- VARYOFF command
 - Command and keyword definitions 459
 - COPY-CMD-DDN 460
 - JCL 461
 - JCL VARYOFF step
 - Sample control statements 461
 - JOURNAL-DDN 460

- VARYOFF command (*continued*)
 - JOURNAL-DSN 460
 - Keyword definitions 459
 - MAX-VOLS-PER-CMD 460
 - SCOPE 460
 - SIMULATE 461
 - SOURCE 460
 - TARGET 460
 - VARY-DDN 460
 - VOL-ALREADY-OFFLINE 461
- VARYOFF command syntax 459
- VARYON 463
- VARYON command
 - Command and keyword definitions 464
 - JCL 466
 - JOURNAL-DDN 464
 - JOURNAL-DSN 464
 - Keyword definitions 464
 - SOURCE 464
 - TARGET 464
 - VARY-DDN 465
- VARYON command syntax 464
- verifying 72
- VOL-ALREADY-OFFLINE
 - VARYOFF command 461
- VOL-ALREADY-ONLINE command
 - VARY-DDN 465
- VOLBKUP-DDN
 - RENAME command 448
- VOLOPTIONS 468
- VOLOPTIONS command
 - CLIP 470
 - Command and keyword definitions 469
 - JCL 471
 - JOURNAL-DDN 469
 - JOURNAL-DSN 469
 - Keyword definitions 469
 - LIST 470
 - NEWTARGETS 471
 - NEWTARGETS-DDN 471
 - NEWTARGETSDEVN 471
 - OFFLINECLIP 470
 - RESUME 470
 - SIMULATE 470
- VOLOPTIONS command syntax 469
- VOLPAIRS
 - COPY command 361
- VOLPAIRS-DDN
 - COPY command 361
 - RESTORE-FROM-DUMPTAPES command 453
- VOLPAIRSDEVN
 - COPY command 361
- VOLPAIRSDEVN-DDN
 - COPY command 361
- VOLPAIRSDEVN-NOCLIP
 - COPY command 361
- VOLPAIRSDEVN-NOCLIP-DDN
 - COPY command 361
- VOLSER mask specification
 - DATA-MOVER(PGM(ADRDSSU)) 368
- volume
 - condition 75
 - volume cloning 535

- volume cloning (*continued*)
 - planning for 75
- Volume copies
 - COPY step JCL 368
- volume copies and renames
 - data set renaming considerations 78
 - migrated application data sets 75
 - RACF and ACF authorities 78
 - requirements 75
 - target ICF catalog renaming
 - considerations
 - populated or empty catalog 78
 - RECATALOG option of RENAME
 - command 78
- volume copy products supported 75
- Volume names
 - For source and target 352
- volumes
 - SMS storage group 75
- volumes not found 368
- VSAM spheres
 - cluster names matching rename
 - mask 75
 - source volume list 75
- VTOCIX_REBUILDER keyword 577

W

- WAIT
 - COPYCHECK 372
 - DB2FIX 383
- WAIT keyword in COPYCHECK
 - step 371
- WAIT-AND-DBD01
 - DB2FIX 384
- what's new 1
- WITHDRAW
 - COPYCHECK 372
- WLM ENVIRONMENT 400
- WORK-DDN
 - DB2GETBACKINFO 386
- WORK-DSN
 - DB2GETBACKINFO 386
- WRONG-VOLSER
 - ONLINECLIP command 432
- WRONG-VOLSER command
 - VARY-DDN 466



Product Number: 5655-N15

Printed in USA

SC19-3493-03

