

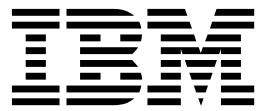
Version 4 Release 3

*IBM DB2 High Performance Unload for
z/OS
User's Guide*



Version 4 Release 3

*IBM DB2 High Performance Unload for
z/OS
User's Guide*



Note:

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

Third Edition (January 2017)

This edition applies to Version 4 Release 3 of DB2 High Performance Unload for z/OS (product number 5655-AA1) and to all subsequent releases and modifications until otherwise indicated in new editions.

© Copyright IBM Corporation 1999, 2017; Copyright Infotel 1999, 2017. 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	v
----------------------------------	---

Chapter 1. DB2 High Performance

Unload overview	1
What's new in DB2 HPU	1
What does DB2 HPU do?	2
DB2 HPU benefits	2
DB2 HPU process and components	3
Tools Customizer overview	3
DB2 HPU and DB2 Tools products scenarios.	6
DB2 HPU compared to the DB2 UNLOAD utility	7
DB2 HPU compatibility.	7
Maximizing performance	8
DB2 HPU terminology	8
Hardware and software requirements	8
Utilities management solutions	9
DB2 HPU documentation and updates.	9
Accessibility features	10

Chapter 2. DB2 HPU customization . . . 13

Authorizing and enabling DB2 HPU	13
Optional: Creating a TSO command for the DB2 HPU interactive component	14
Planning to use Tools Customizer	14
Tools Customizer terminology and data sets	14
Starting and preparing Tools Customizer for use	20
Best Practice: SMP/E and runtime libraries maintenance strategy for Tools Customizer	20
Starting Tools Customizer	24
Modifying Tools Customizer user settings	25
Changing display options	29
Sorting and filtering columns	30
Customizing DB2 HPU	31
Roadmap: Customizing DB2 HPU for the first time	31
Roadmap: Recustomizing DB2 HPU	32
Specifying the metadata library for the product to customize	33
Discovering DB2 HPU information automatically	35
Creating and associating DB2 entries	37
Defining parameters	39
Generating customization jobs	46
Submitting customization jobs	47
Browsing parameters	49
Copying DB2 entries	50
Removing DB2 entries.	51
Deleting DB2 entries	52
Displaying customization jobs	52
Maintaining customization jobs.	53
Using Tools Customizer in a multiple-LPAR environment	53
Optional: Integrating DB2 HPU into DB2 Tools Launchpad	54
Optional: Integrating DB2 HPU into DB2 Administration Tool	55

Chapter 3. Input and output. 57

Input data sources	57
Nonpartitioned table spaces	57
Partitioned table spaces	57
Full image copies or incremental image copies.	58
Last full image copies	60
Output formats	61
Performing physical unloads	61
Performing logical unloads	62
Output encoding scheme	65
Specifying output file options	65
Overriding the automatic output file allocation	65
Specifying one output file per partition	66
Limiting the number of unloaded rows and calling a user exit	67
LOB data processing	68
XML data processing	68
Output data consistency	69
Interaction with the DB2 catalog	69
Consistency considerations	71

Chapter 4. DB2 HPU batch mode operation. 75

Required privileges for running DB2 HPU	75
JCL for running DB2 HPU in batch mode	76
DB2 HPU EXEC statement	76
Processing mode considerations	78
Ddname allocation	79
DB2 HPU-allocated ddnames	79
User-allocated ddnames	80
DB2 HPU syntax	82
PROCESS OPTIONS block syntax and description	83
LISTDEF block syntax and description	84
TEMPLATE block syntax and description	89
GLOBAL OPTIONS block syntax and description	105
UNLOAD block syntax and description	110
OPTIONS block syntax and description	134
SELECT block syntax and description	168
Example: The SYSPRINT data set.	209

Chapter 5. DB2 HPU interactive component 213

DB2 HPU interactive component primary commands	213
Starting the DB2 HPU interactive component in stand-alone mode	215
Starting the DB2 HPU interactive component by using the DB2 Administration Tool	215
Starting DB2 HPU interactive component by using the DB2 Tools Launchpad	217
Specifying general options	218
Including and excluding partitions	221
Specifying COPYDDN file parameters	222

Specifying UNLOAD file parameters	224	DB2 HPU user abend codes	329
Specifying options for the SELECT statement.	225	DB2 HPU messages	333
Updating the list of SELECT statements	228		
Selecting tables.	231	Chapter 8. Reference	395
Processing long names	233	Customization reference	395
Specifying the format of the SELECT statement	234	Discover process parameters	395
Selecting columns for the SELECT statement	238	DB2 HPU library dsnames	395
Sorting columns of a SELECT statement	240	DB2 HPU DB2 parameters	397
Displaying and modifying the USER format of the SELECT statement.	241	DB2 HPU output data parameters	410
Displaying the list of OUTDDN files	245	DB2 HPU DB2 Administration Tool and DB2 Launchpad parameters	438
Displaying and modifying LOADDDN file parameters	248	DB2 HPU conversion parameters.	439
Generating JCL to run a DB2 HPU job	249	DB2 HPU file management parameters.	440
Saving JCL into a file.	250	DB2 HPU reporting parameters	443
Updating the job card	251	DB2 HPU tuning parameters	447
Selecting date formats	251	DB2 HPU sort parameters	453
Selecting time formats	252	Customization templates	455
Selecting timestamp formats	253	INZDB21X member	456
Selecting mask formats	254	Data types for output (TYPE keyword).	458
		Numeric data	458
		Nonnumeric data	459
		DATE, TIME, and TIMESTAMP data	459
		Supported conversions	462
Chapter 6. DB2 HPU user exit	257	Syntax compatibility	463
Register use	257	Fast Unload syntax compatibility.	463
Parameter block	257	UNLOAD PLUS syntax compatibility	469
Sample Assembler user exit	259	DB2 unload compatibility	476
Sample COBOL/2 user exit.	261	Example: Usage of a DB2 UNLOAD JCL with the HPU load module	484
Chapter 7. Troubleshooting	265	Notices	487
Decreasing the amount of necessary memory.	265	Terms and conditions for product documentation	489
Tools Customizer troubleshooting	266		
Gathering diagnostic information.	267	Index	491
Determining the trace data set name	267		
Gathering diagnostic information.	267		
Messages and codes	268		
Tools Customizer messages.	268		

About this information

IBM® DB2® High Performance Unload for z/OS® (DB2 HPU) is a high-speed DB2 utility that unloads DB2 tables from either a table space or from an image copy. This book describes how to customize and use DB2 HPU (in batch and interactively), and includes a description of the messages that DB2 HPU issues.

The following topics are covered in this book:

- Introduction to DB2 HPU functions and states its requirements.
- Description of authorization and configuration of DB2 HPU after installation.
- Description of input to and output from DB2 HPU.
- Description of how to use DB2 HPU in batch mode.
- Description of how to use DB2 HPU interactively.
- Description of the DB2 HPU user exit. You can use this exit to customize your output.
- Description of the user abend codes and messages that are issued by DB2 HPU.
- Description of the use of the TYPE keyword in the SELECT statement.
- Examples of DB2 HPU jobs.
- Syntax of Fast Unload that is compatible with the syntax of DB2 HPU.
- Syntax of UNLOAD PLUS that is compatible with the syntax of DB2 HPU.
- Licensing and trademark information.
- List of related publications and of publications referenced in this book.

The technical changes for this edition are summarized in the Summary of Changes topic. Specific changes since the previous edition of this book are indicated by a vertical bar (|) to the left of a change. Editorial changes that have no technical significance are not noted.

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 High Performance Unload overview

IBM DB2 High Performance Unload (DB2 HPU) is a high-speed DB2 utility for unloading DB2 tables from either a table space or from an image copy. Tables are unloaded to one or more files based on a format that you specify.

Topics:

- “What does DB2 HPU do?” on page 2
- “Utilities management solutions” on page 9
- “DB2 HPU benefits” on page 2
- “DB2 HPU compatibility” on page 7
- “Maximizing performance” on page 8
- “DB2 HPU process and components” on page 3
- “DB2 HPU terminology” on page 8
- “Hardware and software requirements” on page 8
- “DB2 HPU documentation and updates” on page 9
- “Accessibility features” on page 10

What's new in DB2 HPU

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.

SC19-377-02

- The `INLINE`, `INSTREAM_XML_AS_CLOB`, `TRUE_UNICODE`, `AUTO_UNCNT_MAX`, `GBLAPARAL_MAXPART_SET_NULL`, `BY_SQL_ONLY` keywords have been added to the **DB2 HPU additional features (VUU030/ULOPTNS)** parameter.
- The `INZU195I`, `INZU383I`, `INZU308E`, `INZU311E` and `INZU406E` have been created.
- The `INZU184E`, `INZU395E`, `INZI344I` and `INZI345I` messages have been corrected.

SC19-3777-01

- The `AUTO_UNCNT_MAX` and `CHECK_CCSID_STRICT` keywords have been added to the **DB2 HPU additional features (VUU030/ULOPTNS)** parameter.
- The `INZU306E`, `INZU307E`, `INZU308E`, `INZU309E`, and `INZU310E` messages have been added.

SC19-3777-00

- DB2 HPU is enabled with the Tools Customizer component. Use Tools Customizer to customize DB2 HPU after you install it. For more information, see “Tools Customizer overview” on page 3.

What does DB2 HPU do?

DB2 HPU helps you manage and control the unload activity. It works *outside* DB2, directly accessing the VSAM or sequential files that contain the table space or image copy data set.

DB2 HPU also provides superior performance, especially in terms of CPU time and elapsed time. Unless you need to process a complex SQL statement, DB2 HPU typically maximizes performance by reducing processor usage and improves availability by keeping data unavailable for a shorter time.

DB2 HPU can do the following tasks:

- Rapidly unload table spaces
- Parallel processing of several unloads that access the same table space
- Unload against *any* image copy to eliminate interference with DB2 production databases
- Unload selected rows and columns
- Unload every *n* rows and maximum rows
- Generate load control statements for subsequent reload
- Inspect, modify, or discard DB2 rows by using the DB2 HPU user exit
- Create output in multiple formats during a single unload

DB2 HPU benefits

By using DB2 HPU, you can alleviate some of the problems that are associated with unloading large amounts of data.

Large sequential reads of DB2 tables take a long time. Because the large amount of time that large sequential reads of DB2 tables take, the large scans that are required for unload are hard to schedule in the ever-shrinking batch window at DB2 installations.

Performance becomes critical when several unloads read the same table space concurrently. The associated DB2 buffer pool management can cause multiple programs to compete for the same data. This competition between multiple programs can result in the following challenges:

- Conflicts in the use of the DB2 buffer pool
- Writing over buffers that might be serving several unloads
- Multiple reads of the same DB2 pages
- Potential channel conflicts

DB2 HPU relieves these problems. It provides a fast way to sequentially read and share a DB2 table space among multiple unloads. DB2 HPU scans a table space and creates the number of output files that you specify, in the format that you specify. The output format can be any one of the following types:

- DSNTIAUL compatible
- VARIABLE, which lets you quickly create variable-length records
- DELIMITED, which lets you quickly create a delimited file that you can export to another platform
- USER, which lets you specify virtually any type of conversion so that your output appears as you want it

- EXTERNAL, which lets you quickly create fixed length human-readable formatted files

Whenever possible, DB2 HPU processes requests to unload data from the same table space in parallel. You can create different output files during the same unload process at almost no additional cost. For example, you can unload a list of customers who have payments due this week and another list of customers whose birthdays are on the first day of the week. You can create these lists in a single execution of DB2 HPU at a fraction of the cost that is required by traditional dual unload executions.

You can also run DB2 HPU against image copies of the table space, avoiding interference with DB2 production databases.

DB2 HPU process and components

DB2 HPU is a batch utility program that is used to unload DB2 data.

DB2 HPU contains an UNLOAD command and an optional SELECT statement. The SELECT statement syntax is compatible with the DB2 SELECT statement syntax.

By optimizing sequential reads of the table space, DB2 HPU reduces both the elapsed time and the CPU time that is required to process the unloads.

DB2 HPU contains the following components:

- Tools Customizer. You use this component to customize DB2 HPU.
- DB2 HPU interactive component. You can use this component to generate the DB2 HPU batch utility job. Although using this component is not required, inexperienced users might benefit by using it to generate the batch utility job.

Related concepts:

"Tools Customizer overview"

IBM Tools Customizer for z/OS (also referred to as Tools Customizer) standardizes many of the customization processes that are required to customize IBM Tools that run on z/OS.

Chapter 5, "DB2 HPU interactive component," on page 213

The DB2 HPU interactive component can be used to generate the DB2 HPU batch utility job. Although using the interactive application is not required, it can be useful for inexperienced users in generating the batch utility job.

Chapter 4, "DB2 HPU batch mode operation," on page 75

You use the DB2 HPU batch utility program to unload DB2 data.

Tools Customizer overview

IBM Tools Customizer for z/OS (also referred to as Tools Customizer) standardizes many of the customization processes that are required to customize IBM Tools that run on z/OS.

Tools Customizer provides a consistent ISPF interface to ensure that the customization process is the same for all IBM Tools products and solution pack components. It also provides the ability to "discover" parameter values from products or solution pack components that you previously customized manually or by using Tools Customizer.

Features and benefits

Tools Customizer provides the following features:

- A single, consistent ISPF interface ensures that the customization process is the same for all IBM Tools products and solution pack components.
- A Discover EXEC discovers values for common product, LPAR, and DB2 parameters from a product or solution pack component that you previously customized manually or by using Tools Customizer. Each IBM Tools product and solution pack component has a unique Discover EXEC. The discovered parameters are stored in the data store. If the product or solution pack component that you want to customize exists in the Tools Customizer data store, Tools Customizer issues a warning before it overwrites existing values. Use the Discover EXEC by issuing the DISCOVER command on the Customizer Workplace panel.
- The data store retains discovered and manually specified parameter values. Because the parameter information is persistently stored, you have to manually specify or discover parameter values only once. Tools Customizer uses these parameter values where they are applicable.
- A metadata repository contains the members that define the following customization attributes for products and solution pack components:
 - Parameters, tasks, and steps for the product or solution pack component to be customized. Some product or solution pack parameters, tasks, and steps are required.
 - LPAR parameters for the local LPAR. All of the LPAR parameters are required.
 - DB2 parameters for the DB2 subsystem, DB2 group attach name, or DB2 data sharing member on which you will customize the product or solution pack component. All of the DB2 parameters are required.
- Default values are provided for product parameters and solution pack component parameters, LPAR parameters, and DB2 parameters. The default values show examples of how to complete fields.

What's new in Tools Customizer

This topic summarizes the technical changes for this edition. New and changed information is denoted either by a character in the left margin or by blue carets (Symbol that indicates the start of new or changed content) and (Symbol that indicates the end of new or changed content) at the beginning and end of the new or changed information. Editorial changes that have no technical significance are not noted.

- You can create a shared profile so multiple users can customize the same product. For more information, see [Creating a shared profile](#).
- Using the VERIFY/VERIFYOFF command, you can turn on and off parameter verification of product or component parameters.
- Options can be displayed or hidden on certain panels. For more information, see [“Changing display options”](#) on page 29.
- You can now specify specific volumes in which to contain the customization library data set, the discover output data set, and the data store data set.
- You can sort and filter data in panels by column. For more information, see [“Sorting and filtering columns”](#) on page 30.
- Tools Customizer in a multi-LPAR environment is now supported.
- You can now specify the customization library on the Workplace panel.

- Version and Name columns were added to the CCQPHLQ panel where the name field is the name of a product or pack you previously worked with and the version field is the version of the product or pack you previously worked with.

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.

DB2 HPU and DB2 Tools products scenarios

You can use DB2 HPU with some IBM DB2 Tools to quickly unload data.

The following scenarios show how DB2 HPU can enhance the functionality of other DB2 Tools:

DB2 HPU and IBM DB2 Administration Tool for z/OS

DB2 Administration Tool helps you manage DB2 environments efficiently and effectively. When you use the DB2 Admin Alter ALT and Migrate functions, you can use DB2 HPU within a work statement list. Additionally, when you use an ALTER table space redefine against a single table space, you can use DB2 HPU as the unload method.

DB2 HPU and IBM DB2 Object Comparison Tool for z/OS

DB2 Object Comparison Tool compares two sets of DB2 objects. With it, you can compare DB2 objects by creating a batch job in which you specify all aspects of the comparison. When you specify the method for unloading data, you can use DB2 HPU, if it is available.

DB2 HPU and IBM DB2 Object Restore for z/OS

DB2 Object Restore can restore dropped objects and all related dependencies automatically, even if they do not exist in the DB2 catalog. When DB2 Object Restore is configured to access the DB2 HPU product libraries, you can use DB2 HPU to help recover a single table from an image copy data set.

DB2 HPU compared to the DB2 UNLOAD utility

DB2 HPU has some advantages over the DB2 UNLOAD utility.

DB2 HPU uses the following features to simplify the unload process. The DB2 UNLOAD utility provides limited or no support for these features.

SQL support

You can use any SQL statement with DB2 HPU as long as DB2 YES is specified.

Join operations

You can perform join operations. However, performance will be similar to the performance of the DSNTIAUL program.

Multiple output formats

You can create output in multiple formats, such as DELIMITED, VARIABLE, and USER.

User exit

You can use the DB2 HPU user exit to customize the output data set that is created by a SELECT statement. By using the exit, you can examine, modify, or reject DB2 rows.

Image copy processing

You can specify that you want to unload data against either the last full image copy or a different image copy. Additionally, you can unload data from a dropped table by specifying the object ID (OBID). In this case, you must have re-created the table.

DB2 HPU compatibility

To reduce cost and simplify conversion, DB2 HPU offers limited compatibility with the syntaxes that other products use.

DB2 HPU supports the JCL that is used with Fast Unload for DB2, Version 3.1 and the JCL that is used with UNLOAD PLUS for DB2, Version 2.1.01; however, some features of the Fast Unload and UNLOAD PLUS products might be ignored or might be interpreted differently when they are issued by DB2 HPU. In most cases, the amount of work that is required to convert Fast Unload JCL and UNLOAD PLUS JCL to DB2 HPU JCL is reduced.

DB2 HPU supports the Fast Unload and UNLOAD PLUS syntaxes only to the extent that DB2 HPU can perform processing that is like the processing that is

described in the Fast Unload and UNLOAD PLUS syntaxes. Many keywords are ignored, and some options are automatically converted to DB2 HPU syntax.

Maximizing performance

To maximize performance, DB2 HPU uses buffering, synchronization techniques, and parallel processing.

Buffering

When reading data rows, DB2 HPU directly accesses the VSAM clusters that contain the table space. This direct use of VSAM takes maximum advantage of the buffering capability that is provided by VSAM, which means that an entire cylinder can be read with a single I/O.

Synchronization

DB2 HPU permits the parallel execution of several unload jobs that access the same table space; it permits this parallel execution by synchronizing the unloads.

Parallel processing

Unload requests that work on the same table space are processed in parallel whenever possible. Unload requests that work on different table spaces can be processed in parallel if they are specified in the global parallelism setting.

DB2 HPU terminology

DB2 HPU includes several unique terms that you must understand before using DB2 HPU.

logical unload

Uses SELECT statements to enable you to filter the rows and columns that you want to unload and to specify output formats

SQL processing

DB2 HPU passes an SQL statement to DB2 for processing

native processing

DB2 HPU reads the data to unload directly from the data sets where it is stored

physical unload

Starts unloading at the beginning of a table space and does not stop unloading until the entire table space is unloaded

Hardware and software requirements

DB2 HPU requires the same hardware configuration that is required by DB2 for z/OS Version 8 or later.

DB2 HPU requires the following software:

- z/OS Version 1 Release 10 or later
- DB2 Version 9.1 for z/OS or later

Note:

To exploit DB2 10 for z/OS functionality, DB2 HPU V4.2 with appropriate maintenance level (PM78638 for Toleration and PM91909 for exploitation) is required. Previous versions of DB2 HPU cannot exploit DB2 10 for z/OS functionality.

If you want to perform conversions that imply non-SBCS CCSIDs or pairs of SBCS CCSIDs that are not supported by the SYSSTRINGS catalog table, you must install z/OS Support for Unicode.

Utilities management solutions

IBM solutions help IT organizations maximize their investment in DB2 databases while staying on top of some of today's toughest IT challenges.

Today, the size and number of databases continue to grow, and increased data volumes are becoming harder to manage. With an increasing number of applications that need to maintain access to the data, database administrators can struggle to find time to move data when necessary. Extracting and migrating data have become time-consuming and complex processes, which makes it difficult to schedule unloads of large amounts of data. As a database administrator, you might face some of the following challenges:

- Large unload processes that involve sequential reading of DB2 tables
- Slower processes because resources contend with each other for the same data during the unload process

DB2 High Performance Unload is only one of several DB2 Tools products that can help you meet these challenges and achieve higher availability and better performance during data maintenance while enhancing the productivity of both database administrators and system programmers.

The following DB2 tools also provide solutions for managing utilities:

- DB2 Administration Tool
- DB2 Automation Tool
- DB2 Storage Management Utility
- DB2 Utilities Enhancement Tool
- DB2 Utilities Suite

More than ever, the tools that you use for extracting and migrating data can have a critical impact on the efficiency and availability of your IT environment. Reorganization tools from IBM can help with the performance of key functions such as unloading and reloading DB2 data without impacting data access. Designed for ease of use and flexibility, DB2 HPU is a fast and efficient tool to unload and extract data for movement across enterprise systems or for reorganization in-place.

DB2 HPU can be launched from either the DB2 Administration Tool or the DB2 Tools Launchpad. To help you save time and effort, DB2 HPU integrates with DB2 Utilities Suite to offer lists of objects that can be specified by using LISTDEF and TEMPLATE statements.

DB2 HPU documentation and updates

This topic explains where to find DB2 and IMS™ Tools information on the Web, and explains how to receive information updates automatically.

DB2 HPU information on the Web

The DB2 Product publications 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 for z/OS and IMS Tools from the Information Management Software for z/OS Solutions Knowledge Center:

<http://www-01.ibm.com/support/knowledgecenter/SSLTBW/welcome>

Documentation for many DB2 Tools that run on Linux, UNIX, and Windows systems can be found in the IBM DB2 Tools for Linux, UNIX, and Windows Knowledge Center:

<http://www-01.ibm.com/support/knowledgecenter/>

IBM Redbooks publications that cover DB2 and IMS Tools are available from the following Web page:

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

The Data Management Tools Solutions Web site shows how IBM solutions can help IT organizations maximize their investment in DB2 and IMS 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 a weekly e-mail that notifies 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 Support service. You can customize the service so that you receive information about only those IBM products that you specify.

To register with the My Support 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 Support page is displayed, click **add products** to select those products that you want to receive information updates about. The DB2 and IMS Tools category is located under **Software > Data and Information Management > Database Tools & Utilities**.
4. Click **Subscribe to email** to specify the types of updates that you would like to receive.
5. Click **Update** to save your profile.

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 DB2 HPU enable users to:

- 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 Interactive System Productivity Facility (ISPF) User's Guide, Volume 1*, SC34-4822
 - *z/OS TSO/E Primer*, SA22-7787
 - *z/OS TSO/E User's Guide*, SA22-7794

These guides describe how to use ISPF, 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. DB2 HPU customization

The customization process consists of authorizing and enabling DB2 HPU and using Tools Customizer to set parameter values and generate the customization jobs that you will submit.

Prerequisite: Before you can authorize and customize DB2 HPU, you must install DB2 HPU by following the instructions that are provided in the program directory.

Customizing DB2 HPU consists of the following steps:

Table 1. Overview of DB2 HPU customization steps

Step	Description	Page
1	Authorize and enable DB2 HPU.	"Authorizing and enabling DB2 HPU"
2	Optional: Make the DB2 HPU interactive component available through TSO.	"Optional: Creating a TSO command for the DB2 HPU interactive component" on page 14
3	Review the information about planning to Tools Customizer.	"Planning to use Tools Customizer" on page 14
4	Start and prepare Tools Customizer for use.	"Starting and preparing Tools Customizer for use" on page 20
5	Use Tools Customizer to customize DB2 HPU.	"Customizing DB2 HPU" on page 31
6	Optional: Integrate DB2 HPU into DB2 Tools Launchpad.	"Optional: Integrating DB2 HPU into DB2 Tools Launchpad" on page 54
7	Optional: Integrate DB2 HPU into DB2 Administration Tool.	"Optional: Integrating DB2 HPU into DB2 Administration Tool" on page 55

Authorizing and enabling DB2 HPU

Before you can use DB2 HPU for the first time, you must APF authorize the SINZLINK library and enable the DB2 HPU batch utility.

About this task

Enabling the DB2 HPU batch utility lets you unload DB2 data. You must create a batch utility job before you can run the DB2 HPU batch utility.

Procedure

1. Specify the SETPROG command in the following format:

```
SETPROG APF,ADD,DSNAME=SINZLINK-data-set-name,VOLUME=volume-name
```

The following example shows the SETPROG command in the correct format:


```
SETPROG APF,ADD,DSNAME=DMT.PTF420.SINZLINK,VOLUME=V01001
```

You can include this command in a MVS™ procedure, in a job, or in both.

2. Use one of the following methods to enable the DB2 HPU batch utility.

- Put the SINZLINK library in LINKLIST.
- Specify the SINZLINK library in the STEPLIB statement of all jobs. (Be careful not to lose the APF authorizations with the concatenations.)

Related information:

 [MVS System Commands in the z/OS Library Center](#)

Optional: Creating a TSO command for the DB2 HPU interactive component

You can make the DB2 HPU interactive component available as a TSO command.

About this task

With the DB2 HPU interactive component, you can generate the batch utility job to be used to invoke the DB2 HPU batch utility program.

You can run the batch utility program and the DB2 HPU interactive component simultaneously.

Procedure

Use one of the following methods to create a TSO command:

- Concatenate the SINZCLST library to the SYSPROC concatenation.
- Copy the INZHPU procedure into a library in the SYSPROC concatenation.

What to do next

Invoke the DB2 HPU interactive component by issuing the following command:
TSO INZHPU.

Related tasks:

“Starting the DB2 HPU interactive component in stand-alone mode” on page 215
You can start DB2 HPU interactive component in stand-alone mode without using the DB2 Administration Tool or the DB2 Tools Launchpad.

Planning to use Tools Customizer

This topic provides steps to take before using Tools Customizer to customize DB2 HPU.

Procedure

1. Become familiar with Tools Customizer. “Tools Customizer terminology and data sets.”
2. To complete customization, you might need to use the information in the Customization reference. This reference section provides additional information about the product information required to customize DB2 HPU, details of the tasks, steps, and parameters that are displayed on the Product Parameters panel in Tools Customizer, and information about the jobs that will be generated.

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 HPU 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 HPU. If the product to be customized requires DB2 entries, you can customize DB2 HPU only on DB2 entries that are in the associated list. When you customize DB2 HPU, 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 HPU.

Master list

The list of all DB2 entries that are defined but are not associated with DB2 HPU. 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 HPU, 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 HPU, 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 HPU. These parameters are defined by DB2 HPU and are stored in a data member that is defined by DB2 HPU.

LPAR parameters

Parameters on the local LPAR that are required to customize DB2 HPU. These parameters are defined by Tools Customizer and are stored in an LPAR parameter data member.

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 HPU, all of the required parameters must be defined.

If required parameters for the the product parameters, LPAR 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, LPAR parameters, and DB2 parameters for the DB2 entries on which DB2 HPU will be customized.

The following table shows the meaning of the status types. Each status is defined differently for each type of parameter.

Table 2. Status types for the product, the LPAR, and the DB2 entries

Status	Product	LPAR	DB2 entries
Incomplete	The required product parameters are not defined.	The required parameters are not defined.	The required parameters are not defined.

Table 2. Status types for the product, the LPAR, and the DB2 entries (continued)

Status	Product	LPAR	DB2 entries
Discovered	The product parameter definitions were discovered by using the product Discover EXEC.	N/A	N/A
Ready to Customize	The required product, 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.
Verify Values	The required product or component parameter values are defined but they either have not been verified or verification is not enabled on the Product or Component Parameters panel.	The required LPAR parameter values are defined, but they either have not been verified or verification is not enabled on the LPAR Parameters panel.	The required DB2 parameter values are defined, but they either have not been verified or verification is not enabled on the DB2 Parameters panel.
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 37

You can create new DB2 entries and associate them with DB2 HPU.

“Copying DB2 entries” on page 50

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

“Removing DB2 entries” on page 51

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 HPU with Tools Customizer. These data sets are supplied by DB2 HPU, supplied by Tools Customizer, or allocated by Tools Customizer.

DB2 HPU 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*.SINZDENU, 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 HPU Discover EXEC. When you customize DB2 HPU, 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 HPU, the lowest-level qualifier is SINZDENU. 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 HPU. 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 HPU Discover EXEC. The DB2 HPU Discover EXEC retrieves the metadata and values for the parameters from a previous customization of DB2 HPU.

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 HPU.

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 HPU, 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 3. 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

Table 3. Data set attributes for allocating the Discover output, data store, and customization library data sets (continued)

Data set	Organization	Record format	Record length	Block size	Data set name type
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.

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 HPU.

Best Practice: SMP/E and runtime libraries maintenance strategy for Tools Customizer

Tools Customizer creates relationships between the values for the Product Parameters, LPAR Parameters, and DB2 Subsystem Parameters for each Tools Customizer enabled product. Determining the correct maintenance strategy for your Tools Customizer runtime libraries, after SMP/E processing, can reduce problems working with Tools Customizer and the enabled products through their life cycles.

Tools Customizer has very specific requirements for data set names:

- Only one DATASTOR data set exists per LPAR
- The product metadata library data set names do not change during the life of that release of the Tools Customizer enabled product.

The DATASTOR data set is the repository for all the information that Tools Customizer requires to generate customization JCL for enabled products.

When you update and save the Tools Customizer Settings panel (CCQPSET), as described in “Modifying Tools Customizer user settings” on page 25, the name of the DATASTOR data set is saved in the ISPF profile. This allows Tools Customizer to know the active DATASTOR data set when the TSO user id logs in and starts the Tools Customizer EXEC.

Maintenance scenarios

IBM expects maintenance to be applied to libraries which are then used by Tools Customizer. In practice, different customer shops distribute SMP/E APPLY maintenance in different ways.

The following scenarios explains some considerations and alternatives for determining your maintenance strategy. The one overriding objective is to preserve and maintain the same data set names for the Tools Customizer instance.

|
| **Apply SMP/E maintenance to the same data sets (using the SMP/E APPLY command):**

| If you apply SMP/E maintenance (using the SMP/E APPLY command)
| using the same data set names with each maintenance cycle, you can either
| use these target libraries as your Tools Customizer runtime libraries or you
| can copy the SMP/E target data sets to the runtime libraries that are used
| by Tools Customizer to customize enabled products.

| Tools Customizer assumes that if the product metadata library (*DENU)
| has the same name, this metadata library is for the same release of the
| enabled product. For example, assume that you customize DB2 Log
| Analysis Tool v3.3 and name the metadata library SYS2.DB2T00L.SALADENU,
| with no indication of the version or release. You then upgrade to DB2 Log
| Analysis Tool v3.4 and employ the same naming convention,
| SYS2.DB2T00L.SALADENU. Tools Customizer will assume that you are
| continuing to work with DB2 Log Analysis Tool v3.3 and will report v3.3
| on panels and continue to use the same v3.3 Customization Library data
| set.

| A more sustainable approach to naming the data sets is to include a
| product version, release identifier, or other distinguishing qualifier in the
| name of the metadata library, so that Tools Customizer can determine the
| new product release when you upgrade. For example, using metadata
| library names that include a product version, release identifier, or other
| distinguishing qualifier, similar to the following, can make product
| maintenance and upgrades easier:

- | • DB2T00L.R330.SALADENU for DB2 Log Analysis Tool v3.3
- | • DB2T00L.R340.SALADENU for DB2 Log Analysis Tool v3.4

| **Apply SMP/E maintenance to new data sets (using the SMP/E APPLY command):**

| If you apply SMP/E maintenance (using the SMP/E APPLY command) to
| new data sets rather than to the same data sets, the next time you open the
| product metadata library, Tools Customizer will return a data set error that
| indicates that the library name is being used by another product or
| component.

| For example, assume that you name the DB2 High Performance Unload for
| z/OS target metadata libraries to reflect the date of an upgrade or to reflect
| a specific RSU, as follows:

- | • To reflect a specific upgrade date (August 2014):
 - | – DB2T00L.PTF420.SINZDBRM.D201408
 - | – DB2T00L.PTF420.SINZDENU.D201408
 - | – DB2T00L.PTF420.SINZLOAD.D201408
- | • To reflect a specific RSU (RSU 1406):
 - | – DB2T00L.PTF420.SINZDBRM.RSU1406
 - | – DB2T00L.PTF420.SINZDENU.RSU1406
 - | – DB2T00L.PTF420.SINZLOAD.RSU1406

| Using either of these naming conventions, the next time you start the Tools
| Customizer EXEC, it will return a data set error.

| To handle this type of SMP/E maintenance processing, you can do either
| of the following:

- | • Define aliases (using ALIAS control statements) to reference the
| appropriate libraries for Tools Customizer processing.

- Copy the SMP/E libraries to a set of runtime libraries that are specifically for Tools Customizer processing.

Define aliases (using ALIAS control statements) to reference the appropriate libraries for Tools Customizer processing:

Defining aliases (using ALIAS control statements) for the SMP/E created new product data set names is likely the best strategy when planning for Tools Customizer.

If you are setting up Tools Customizer for the first time, consider specifying the product library data set names with an indicator that these data sets will be used for Tools Customizer processing. For example, use data set names similar to the following names:

- TCZ.PTF420.SINZDBRM
- TCZ.PTF420.SINZDENU
- TCZ.PTF420.SINZLOAD

After applying maintenance using SMP/E, which creates new product library data sets, you should define aliases (using ALIAS control statements) for the new data set names to the data set names that Tools Customizer originally processed. For example, the following maintenance data sets have aliases defined to the original data sets:

- DB2T00L.PTF420.SINZDBRM.RSU1406 --> TCZ.PTF420.SINZDBRM
- DB2T00L.PTF420.SINZDENU.RSU1406 --> TCZ.PTF420.SINZDENU
- DB2T00L.PTF420.SINZLOAD.RSU1406 --> TCZ.PTF420.SINZLOAD

You will need to define an alias (using ALIAS control statements) to each of the following IBM-distributed Tools Customizer data sets:

- SCCQDENU
- SCCQEXEC
- SCCQLOAD
- SCCQMENU
- SCCQPENU
- SCCQSAMP
- SCCQTENU

If you have already set up Tools Customizer and customized the product, you will have to define aliases (using ALIAS control statements) for the newly created data set names to the data sets that were specified when the product was originally customized using Tools Customizer.

After defining the aliases, you should be able to run Tools Customizer successfully.

Note: Only define aliases for IBM-distributed SMP/E libraries.

Restriction: Do not define an alias for any Tools Customizer created data sets, like the following three data sets on the Tools Customizer Settings panel (CCQPSET):

- Customization library qualifier
- Discover output data set
- Data store data set

Do not define an alias for any data sets that you create on behalf of a product, for example the DB2 High Performance Unload for z/OS parameter library (INFPLIB).

Copy the SMP/E libraries to a set of runtime libraries that are specifically for Tools Customizer processing:

If you are setting up Tools Customizer for the first time, consider specifying the product library data set names with an indicator that these data sets will be used for Tools Customizer processing. For example, use data set names similar to the following names:

- TCZ.R420.SINZDBRM
- TCZ.R420.SINZDENU
- TCZ.R420.SINZLOAD

After applying maintenance using SMP/E, which creates new product library data sets, you should copy the new data sets to the data sets that Tools Customizer originally processed. For example, copy the following maintenance data sets to the original data sets:

- DB2T00L.PTF420.SINZDBRM.RSU1406 --> TCZ.R420.SINZDBRM
- DB2T00L.PTF420.SINZDENU.RSU1406 --> TCZ.R420.SINZDENU
- DB2T00L.PTF420.SINZLOAD.RSU1406 --> TCZ.R420.SINZLOAD

You will need to copy each of the following IBM-distributed Tools Customizer data sets:

- SCCQDENU
- SCCQEXEC
- SCCQLOAD
- SCCQMENU
- SCCQPENU
- SCCQSAMP
- SCCQTENU

If you have already set up Tools Customizer and customized the product, you will have to copy the newly created data sets to the data sets that were specified when the product was originally customized using Tools Customizer).

After copying the SMP/E data sets to the Tools Customizer instance libraries, you should be able to run Tools Customizer successfully.

Note: Only copy IBM-distributed SMP/E libraries.

Restriction: Do not copy any Tools Customizer created data sets, like the following three data sets on the Tools Customizer Settings panel (CCQPSET):

- Customization library qualifier
- Discover output data set
- Data store data set

Do not copy any data sets that you create on behalf of a product, for example the DB2 High Performance Unload for z/OS parameter library (INFPLIB).

Related tasks:

Modifying Tools Customizer user settings

Before you can customize a product or a component with Tools Customizer, you must review the settings that Tools Customizer uses.

Related information:

The SMP/E APPLY command

The APPLY command specifies which of the received SYSMODs are to be selected for installation in the target libraries.

Alias processing: SMP/E for z/OS Commands

When an element with aliases is processed, both the element and its aliases are updated. SMP/E does not check the aliases against elements maintained in the target zone.

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'

| You can also specify a trace data set name and a user profile when you run
| the REXX EXEC.

- The default trace data set name is *USERID.CCQ.TRACE*. To specify a different trace data set name, append the trace data set name to the command. For example, to specify a trace data set name of *CCQTCZ.MYTRACE*, enter:

| EX 'CCQTCZ.USABSAND.SCCQEXEC(CCQTCZ)' 'CCQTCZ.USABSAND, CCQTCZ.MYTRACE'

- To specify a user profile other than your own, append the user profile name to the command. For example, to specify a user profile of *SHRPROF*, enter:

| CCQTCZ.USABSAND.SCCQEXEC(CCQTCZ)' 'CCQTCZ.USABSAND, ,CCQTCZ.SHRPROF'

| Tools Customizer will use the settings from the specified profile. This
| profile will be updated when you exit Tools Customizer, but your own
| profile will remain unchanged.

| **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 HPU.

| **Modifying Tools Customizer user settings**

| Before you can customize DB2 HPU 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 HPU 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  OPTIONS

Product Customization Settings

*Customization library qualifier . . DB2TOOL.PRODUCT.CUST
  Volume serial . . . . .
*Use DB2 group attach . . . . . YES (YES/NO)

Tools Customizer Library Settings

*Metadata library . . . . . DB2TOOL.CCQ110.SCCQDENU
*Discover output data set . DB2TOOL.CCQ110.DISCOVER
  Volume serial . . . . .
*Data store data set . . . . . DB2TOOL.CCQ110.DATAST01
  Volume serial . . . . .

User Job Card Settings for Customization Jobs

==>> //          JOB (ACCOUNT),'NAME',

```

Figure 1. The Tools Customizer Settings panel (CCQPSET)

Note: An asterisk next to a field indicates that the field is required.

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 HPU 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 (').

Volume serial

The volume name in which the customization library will reside. If you don't specify a volume name, it will be assigned by the system.

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 4. 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:

- DB2C, which is a stand-alone DB2 subsystem
- DB2A, 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
DB2C  --      910  NFM  SYSADM  2010/11/09  Ready to Customize
DB2A  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 5. 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
DB2C	--	Yes	SSID
		No	SSID
DB2A	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 and LPAR parameters. The parameters that are displayed on the LPAR Parameters panel and 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 HPU Discover EXEC is stored. Each product has its own Discover EXEC. The Discover EXEC retrieves the product, LPAR, 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 (').

Volume serial

The volume name in which the discover output data set will reside. If you don't specify a volume name, it will be assigned by the system.

Data store data set

The name of the data set where Tools Customizer stores information about product, LPAR, 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 (').

Volume serial

The volume name in which the data store data set will reside. If you don't specify a volume name, it will be assigned by the system.

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 HPU or to change parameter settings.

Related concepts:

“Customizing DB2 HPU” on page 31

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

Changing display options

You can choose which types of information to show on Tools Customizer panels. You can also copy your user profile to another data set so that it can be shared with other users.

About this task

By using the `OPTIONS` command, you can choose to show or hide the following information on Tools Customizer panels:

- The instructions on all panels
- The Product to Customize section on the Customizer Workplace panel (CCQPWRK)
- The Usage Notes section on the Product Parameters panel (CCQPPRD)

The `OPTIONS` command also allows you to copy your user profile to another data set so that it can be shared with other users. By sharing a copy of your profile, other users can customize the same products that you initially customized or started to customize.

Procedure

1. On any Tools Customizer panel, issue the `OPTIONS` command. The Miscellaneous Options panel (CCQPOPT) is displayed, as shown in the following figure. By default, all panel display options are preselected with a slash (/), which means that they will be displayed.

```

CCQPOP1                               Miscellaneous Options
Command ==>>> _____

Enter a / to select options and press Enter. To cancel, press End.

Panel Display Options
 / Show the panel instructions
 / Show the Product to Customize section
 / Show the Usage Notes section

User Profile Sharing Option
_ Copy user profile to another data set

```

Figure 2. The Panel Display Options panel (CCQPOPT)

2. To hide the panel instructions, the Product to Customize section, or the Usage Notes section, remove the slash from the appropriate option or options.
3. To copy your user profile to another data set so that it can be shared with other users:
 - a. Type a slash in the **Copy user profile to another data set** field and press Enter.
 - b. Specify the fully qualified name of the data set into which you want to copy the current user profile. If the data set name exceeds 42 characters, enclose the name in quotation marks. ALTER or UPDATE authorization to this data set is required.
 - c. Optionally specify a volume name in which the user profile data set will reside. If you don't specify a volume name, it will be assigned by the system.
4. Press Enter to save your changes.

Sorting and filtering columns

You can sort data in Tools Customizer columns by up to two columns. You can also filter the data in columns to display only the data that matches the filter criteria that you specify.

About this task

Sorting and filtering is available only on the Customizer Workplace panel, the Finish Product Customization panel, the Associate DB2 Entry for Product panel, and the Copy Associated DB2 Entry panel.

Procedure

The following instructions describe how to sort and filter data in Tools Customizer columns:

- To sort data in Tools Customizer columns, issue the SORT command.

On the SORT command, specify up to two column names followed by the sort order: A for ascending or D for descending. If you don't specify a sort order, the default sort order is used, which can change depending on the column type.

For example, the following command sorts the column entries by SSID in ascending order, and then by GrpAttch in descending order within SSID.

```
COMMAND ==>>> SORT SSID A GrpAttch D
```

You cannot specify the **Cmd** column on the SORT command.
- To filter data in Tools Customizer columns, overwrite the asterisk (*) under the column names with the filtering arguments for those columns.

For example, to filter SSIDs that start with DB, overwrite the *\ under the SSID column with DB or DB*. When you press Enter, all the SSIDS that meet that criteria, such as DB01 and DB02, are displayed.

A filter argument in the form DB* means that only the characters up to the asterisk are considered. When you specify an asterisk in the last nonblank position of the argument, asterisks embedded in the argument are treated as data.

Customizing DB2 HPU

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

Customization roadmaps describe the steps that you must complete to customize DB2 HPU. 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 6. Customization roadmaps

Environment description	Roadmap
You do not have a customized version of DB2 HPU, and you need to customize it for the first time.	"Roadmap: Customizing DB2 HPU for the first time"
You have a customized version of of DB2 HPU, but you want to change one or more parameter values.	"Roadmap: Recustomizing DB2 HPU" on page 32

Roadmap: Customizing DB2 HPU for the first time

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

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.
- The LPAR ISPF libraries that are required to submit the jobs are known.
- 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 HPU for the first time.

Table 7. Steps for customizing DB2 HPU 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.SINZDENU</i> .	"Specifying the metadata library for the product to customize" on page 33
2	Create new DB2 entries and associate them with DB2 HPU.	"Creating and associating DB2 entries" on page 37

Table 7. Steps for customizing DB2 HPU for the first time (continued)

Step	Description	Instructions
3	Define the required parameters.	"Defining parameters" on page 39
4	Generate the customization jobs for the product or for the DB2 entries on which DB2 HPU is ready to be customized.	"Generating customization jobs" on page 46
5	Submit the generated customization jobs.	"Submitting customization jobs" on page 47

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

Table 8. Administrative tasks

Description	Instructions
Browse the different types of parameters.	"Browsing parameters" on page 49
Copy an existing DB2 entry to the list of DB2 entries on which DB2 HPU can be customized.	"Copying DB2 entries" on page 50
Remove one or more DB2 entries from the associated list.	"Removing DB2 entries" on page 51
Delete one or more DB2 entries from the master list.	"Deleting DB2 entries" on page 52
Display a list of customization jobs that have been previously generated.	"Displaying customization jobs" on page 52
Maintain the customization jobs in the customization library.	"Maintaining customization jobs" on page 53

Roadmap: Recustomizing DB2 HPU

This roadmap lists and describes the steps to change parameter values and regenerate customization jobs for DB2 HPU 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 HPU.

Table 9. Required steps for recustomizing DB2 HPU

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.SINZDENU</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 HPU parameters” on page 39 • “Defining LPAR parameters” on page 41 • “Defining DB2 parameters” on page 44
3	Generate the customization jobs for the product or for the DB2 entries on which DB2 HPU is ready to be customized.	“Generating customization jobs” on page 46
4	Submit the new generated customization jobs.	“Submitting customization jobs” on page 47

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

Table 10. Administrative tasks

Description	Instructions
Browse the different types of parameters.	“Browsing parameters” on page 49
Copy an existing DB2 entry to the list of DB2 entries on which DB2 HPU can be customized.	“Copying DB2 entries” on page 50
Remove one or more DB2 entries from the associated list.	“Removing DB2 entries” on page 51
Delete one or more DB2 entries from the master list.	“Deleting DB2 entries” on page 52
Display a list of customization jobs that have been previously generated.	“Displaying customization jobs” on page 52
Maintain the customization jobs in the customization library.	“Maintaining customization jobs” on page 53

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 HPU. This information controls what is displayed on the Product Parameters panel, the LPAR Parameters panel, and the DB2 Parameters panel.

After DB2 HPU has been SMP/E installed, the default name of the product metadata library is *high_level_qualifier.SINZDENU*, 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 Product or Pack 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:

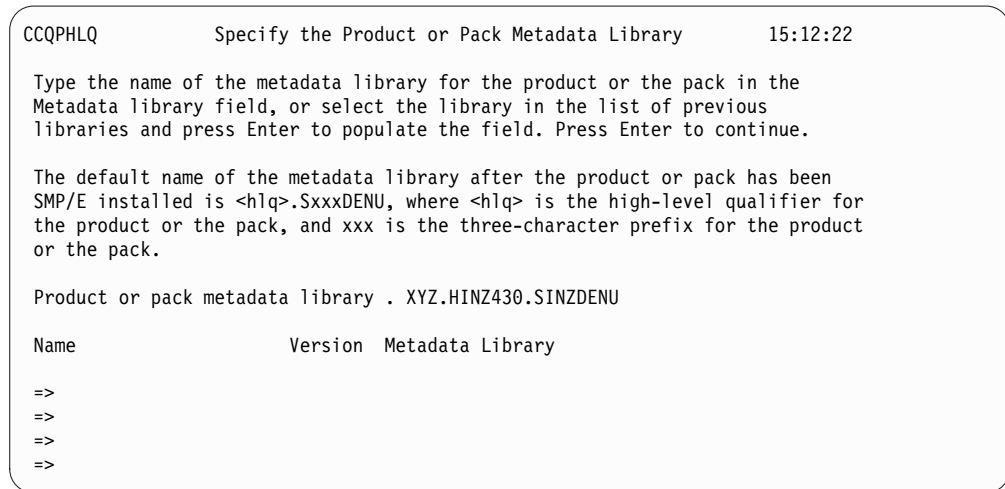


Figure 3. The Specify the Metadata Library panel

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 **Product or pack 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 HPU for the first time, you must use this method.
 - Place the cursor in any column of the Recent Metadata Libraries list, and press Enter to populate **Product or pack metadata library** field. Press Enter again to select product or pack for customization.

Results

If you are customizing DB2 HPU 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 HPU for the first time

Do not run the DB2 HPU 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 HPU parameters.

Customizing DB2 HPU from a previous or current customization

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

Discovering DB2 HPU information automatically

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

About this task

Tip: Using the DB2 HPU 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 HPU provides the Discover EXEC that you will run. Therefore, the information that can be discovered depends on DB2 HPU.

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 HPU 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                19:57:11

For the product you are customizing, the Discover EXEC retrieves product
information from an already customized product. Specify the required
information. To save your information and run the Discover EXEC, issue the RUN
command. To save your information and stay on this panel, issue the SAVE
command. To verify the syntax of your information without saving it, press
Enter. To save and exit, press End.

Commands: RUN  SAVE

Product to Customize
  Product metadata library : XYZ.HINZ430.SINZDENU > LPAR. . . : ZPS1
  Product name . . . . . : IBM DB2 High Performance U > Version . . : 4.3.0

Discover EXEC for Extracting Information from an Already Customized Product
*Discover EXEC library . . . XYZ.HINZ430.SINZCLST
Discover EXEC name . . . . : INZTDSC
*Discover output data set . . XYZ.HINZ430.DISCOUT

Information for Discover EXEC
Name of the PARMLIB library of a previous version (4.2.0 and later)
                                XYZ.HINZ420.SINZRSAV >
Previous data set of variables to be retrieved (4.1.0 and earlier)
>
Previous data set with customized file dsnames (4.1.0 and earlier)
>

Verbose mode for procedure . . . . . OFF (OFF,ON)

```

Figure 4. 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 HPU Discover EXEC.
 - Discover EXEC name**
The name of the DB2 HPU Discover EXEC.
 - Discover output data set**
The fully qualified data set where output from the DB2 HPU 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 HPU. They show the information that is required to run the DB2 HPU Discover EXEC.
4. Issue the RUN command to run the DB2 HPU Discover EXEC. Alternatively, save your information without running the DB2 HPU Discover EXEC by issuing the SAVE command. If you issue the RUN command to run the DB2 HPU Discover EXEC, the parameter information is discovered for DB2 HPU, and the Customizer Workplace panel is displayed.

Results

The discovered parameter values for DB2 HPU 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 HPU on new DB2 entries, create and associate the entries.
- If DB2 entries were discovered and you want to customize DB2 HPU on only these entries, define the parameters.

Related tasks:

“Creating and associating DB2 entries”

You can create new DB2 entries and associate them with DB2 HPU.

“Defining parameters” on page 39

To customize DB2 HPU, you must define DB2 HPU parameters, LPAR 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 HPU.

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:

```

CCQPDAD                Associate DB2 Entry for Product                16:36:42
Command ==>>>                Scroll ==>> PAGE

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 new DB2 entries

DB2 Entries
Line commands: A - Associate C - Copy D - Delete
Cmd SSID GrpAttch
* *
----- End of DB2 entries -----

```

Figure 5. 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 to create one DB2 entry, or issue CREATE *nm* to create multiple DB2 entries, where *nm* is the number of new entries to be created. The Create DB2 Entries panel is displayed, as shown in the following figure:

```

CCQPCDB                               Create DB2 Entries                               Row 1 of 1
Command ==>>>                               Scroll ==>> PAGE

Specify the SSID, the group attach name, or both in the appropriate columns
for each new DB2 entry and press Enter. To create additional entries, issue
the Inn line command, where nn is the number of entries to be inserted, and
press Enter. To cancel, press End.

New DB2 Entries
Line commands: I - Insert into list   R - Remove from list
Cmd  SSID  GrpAttch  Message
----- End of DB2 entries -----

```

Figure 6. The Create DB2 Entries 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:

```

CCQPDA D                               Associate DB2 Entry for Product                               Row 1 to 3 of 3
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 new DB2 entries

DB2 Entries
Line commands: A - Associate  C - Copy  D - Delete
Cmd  SSID  GrpAttch
*      *
DBAA  --
DBAB  --
DBAC  --
----- End of DB2 entries -----

```

Figure 7. Associate DB2 Entry for Product panel

- 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 HPU, 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 HPU.

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 15

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

Defining parameters

To customize DB2 HPU, you must define DB2 HPU parameters, LPAR parameters, and DB2 parameters, if your customization requires DB2 entries.

About this task

You must define the DB2 HPU parameters first for the following reasons:

- If you ran the DB2 HPU 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 HPU, 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 HPU parameters

DB2 HPU parameters are specific to DB2 HPU.

About this task

If you ran the DB2 HPU 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 HPU parameters, a note labeled **Important** will display the correct sequence on this panel.

```

CCQPPRD                Product Parameters                20:06:24

Complete the following tasks to customize the products. The required tasks,
required steps within a required or selected task, and required parameters
are preceded by an asterisk (*). Ensure that values are specified for the
required parameters. Press End to save and exit.

Commands: SAVE  VERIFYOFF
Line Commands: / - Select

Product to Customize
  Product metadata library : XYZ.HINZ420.SINZDENU  > LPAR. . . : ZPS1
  Product name . . . . . : IBM DB2 High Performance U > Version . : 4.3.0

Product customization library : XYZ.HINZ4.3.0.$ZPS1$.INZ4.3.0
                                                                    More:  +

IMPORTANT:
Refer to DB2 High Performance Unload V4.3 User's guide for additional
information about the setting of the variables.

Required parameters
Application plan for DB2 HPU (VUM011/PLAN0BJT)
                                                                    PLN4570P
Package collection for DB2 HPU (VUM030/COLLOBJT)
                                                                    COLL4570

/ Configuration of DB2 HPU

/ Specify values for variables
  Quiesce of SYSDBASE and DBD01 for the batch utilities (VUM014/QUIESCAT)
                                                                    YES  (YES,NO,OFF,FORCE)
  BUFNO for sequential QSAM (VUM022/QSBUFNO)
                                                                    60
  BUFND for sequential VSAM (VUM023/VSBUFND)
                                                                    360
  Number of records in sort work area (VUM024/SRTNBVRE)
                                                                    800
  Minimum size in bytes for sort work area (VUM025/SRTVSMIN)
                                                                    1000000

```

Figure 8. The Product Parameters panel

You can use the following primary commands on this panel:

SAVE Saves the specified product or component parameter values.

VERIFY / VERIFYOFF

Use the VERIFY and VERIFYOFF commands to turn on and off parameter verification of product or component parameters. Before you can generate customization jobs, you must verify that all required parameters are set to a valid value. The product or component parameter status of Verify Values on the Customize Workplace panel indicates that the values have not been verified.

Enter these commands either by typing them in the command field and pressing Enter or by positioning the cursor on the command and pressing Enter. When VERIFY is active, VERIFYOFF is displayed so that you can toggle between the two states. By default, verification is turned on when you display the DB2 Parameters panel, and the verification state is reset to VERIFY every time you exit the Customization Workplace panel by pressing PF3.

Turning verification off is useful when you need to exit the panel before you have entered all of the required parameters, but you want to save the parameters that you have specified. When you disable verification, it is disabled only for the Product or Component Parameters Values 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, Verify Values, or Discovered, edit these parameters.

Related tasks:

“Defining LPAR parameters”

LPAR parameters are parameters on the local LPAR that are required to customize DB2 HPU.

“Defining DB2 parameters” on page 44

DB2 parameters are parameters for a DB2 entry.

Defining LPAR parameters

LPAR parameters are parameters on the local LPAR that are required to customize DB2 HPU.

Procedure

1. Specify E next to the **LPAR parameters** field, and press Enter. The LPAR Parameters panel is displayed, as shown in the following figure:

```

CCQPLPR                                LPAR Parameters                                20:16:07

Browse the values for all of the LPAR parameters. Press End to exit.

Product to Customize
  Product metadata library : XYZ.HINZ430.SINZDENU > LPAR. . . : ZPS1
  Product name . . . . . : IBM DB2 High Performance U > Version . . : 4.3.0

                                                                    More:    +

Other Parameters
  Unit name for batch work data sets . . . : WORK

DB2 HPU Library dsnames
  Load modules library (SINZLOAD) . . . . : XYZ.HINZ430.MVSLoad    >
  APF load modules library (SINZLINK) . . : XYZ.HINZ430.MVSLINK   >
  Samples library (SINZSAMP) . . . . . : XYZ.HINZ430.INFSAMP     >
  PARMLIB library (INZPLIB/INFPLIB) . . . : XYZ.HINZ430.INFPLIB    >
  Product's DBRM library (SINZDBRM) . . . : XYZ.HINZ430.DB2DBRM     >
  REXX EXEC library (SINZCLST) . . . . . : XYZ.HINZ430.ISPCLIST   >
  ISPF load modules library (SINZLLIB) . . : XYZ.HINZ430.ISPLOAD    >
  ISPF messages library (SINZMLIB) . . . . : XYZ.HINZ430.ISPMSGS   >
  ISPF panels library (SINZPLIB) . . . . . : XYZ.HINZ430.ISPPANEL  >
  ISPF skeletons library (SINZSLIB) . . . . : XYZ.HINZ430.ISPSKELS   >
  ISPF tables library (SINZTLIB) . . . . . : XYZ.HINZ430.ISPTBL     >

DB2 HPU File management parameters
  Volume name of migrated object (VZM005/INFVSMIG)
                                     MIGRAT
  Assignable devices number per tape unit/storage class (VZM010/TAPEUNIT) >
  Maximum number of disk units to allocate a work file (VUX010/LIMUNIT)
                                     9
  Maximum size for primary allocation of a work data set (VUX019/WRKMXPR)
                                     500000
  DSCB model for allocation of GDS (VUX033/TMPLDSCB)
                                     >
  Volume(s) for allocation of temporary data sets (VUM018/WRKVOL)

```

Figure 9. The LPAR Parameters panel

You can use the following primary commands on this panel:

SAVE Saves the specified product or component parameter values.

VERIFY / VERIFYOFF

Use the VERIFY and VERIFYOFF commands to turn on and off parameter verification of LPAR parameters. Before you can generate customization jobs, you must verify that all required parameters are set to a valid value. The LPAR parameter status of Verify Values on the Customize Workplace panel indicates that the values have not been verified.

Enter these commands either by typing them in the command field and pressing Enter or by positioning the cursor on the command and pressing Enter. When VERIFY is active, VERIFYOFF is displayed so that you can toggle between the two states. By default, verification is turned on when you display the LPAR Parameters panel, and the verification state is reset to VERIFY every time you exit the Customization Workplace panel by pressing PF3.

Turning verification off is useful when you need to exit the panel before you have entered all of the required parameters, but you want to save the parameters that you have specified. When you disable verification, it is disabled only for the LPAR Parameters Values panel.

- |
- | 2. Specify values for all required parameters that are displayed. Many 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.

| The following LPAR parameters can contain 1 - 64 values:

- LPAR macro library
- Message library
- Panel library
- Skeleton library
- ISPF table input library
- ISPF user profile library
- File tailoring output library
- Link list library
- Command procedures library
- Macro library
- Link-edit library
- Load library
- Started task library name

- |
- | 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 Customizer Workplace panel is displayed, and the status of the LPAR

| parameters is Ready to Customize.

|

| **What to do next**

| If the status of other parameters on the Customizer Workplace panel is Incomplete,

| Verify Values, or Discovered, edit these parameters.

|

| **Related tasks:**

| “Defining DB2 HPU parameters” on page 39

| DB2 HPU parameters are specific to DB2 HPU.

| “Defining DB2 parameters” on page 44

| 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 HPU 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 37.

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                13:36:01
Ensure that values are specified for the required DB2 parameters. Press End
to save and exit.

Commands: SAVE  VERIFYOFF                                More:    +

DB2 subsystem ID . . . . . : DB01
Group attach name . . . . . :

General DB2 Information - common
*Mode . . . . . NFM (CM,CM8,CM9,NFM)
*Level number . . . . . 101 (810,910,101,111)

DB2 Libraries - common
*Load library . . . . . DSN.SDSNLOAD                Add
*Run library . . . . . DSN.RUNLIB.LOAD              Add
*Exit library . . . . . DSN.SDSNEXIT                Add

DB2 Buffer Pools - common
*Name of the 4 KB buffer pool . . . . . BP0

DB2 Utilities - common
*Plan name for the DSNTIAD utility . . . . . DSNTIAD

XYZ DB2 Parameters
Plan for DB2 XYZ Tool . . . . . SYSTOOLS
Database for XYZ objects . . . . . DLCDB
XYZ storage group . . . . . SYSTOOLS
Drop repository database first . . . . . D (Y, N, D)
```

Figure 10. The DB2 Parameters panel

```

CCQPDB2                DB2 Parameters                Top of data

Enter values for all of the DB2 parameters. Press End to exit.

Product to Customize
  Product metadata library : XYZ.HINZ430.SINZDENU > LPAR. . . : ZPS1
  Product name . . . . . : IBM DB2 High Performance U > Version . . : 4.3.0

                                                    More:    +

DB2 subsystem ID . . . . . : DZ7M
Group attach name . . . . . :

General DB2 Information
Mode . . . . . : CM (CM,CM8,CM9,NFM)
Level Number . . . . . : (,910,101,111)
Mixed Mode . . . . . : YES (YES,NO)

DB2 Libraries
Load Library . . . . . : DSN.SDSNLOAD > Add...
Run Library . . . . . : DSN.RUNLIB.LOAD > Add...
Exit Library . . . . . : DSN.SDSNEXIT > Add...

DB2 Utilities
SYSAFF for DB2 utilities . . . . . : MVS1
Plan name for the DSNTIAD utility . . . : DSNTIA71

DB2 HPU DB2 Parameters
Owner of the plan created for DB2 HPU (VUM012/PLANOWN)
      POLETEST
Grant on the plan created for DB2 HPU (VUX011)
      PUBLIC >
User used to quiesce the catalog tablespaces (VUM020/QUIESUSR)
      INSTALL_SYSOPR
User used to quiesce the tablespace to be unloaded (VUM031/QUITUSR)
      INSTALL_SYSOPR
User used when issuing SELECT using dynamic SQL (VUM032/ACTLGUSR)
      INSTALL_SYSADM
User who runs the DISPLAY command (VUM028/DISPLUSR)

```

Figure 11. The DB2 Parameters panel

You can use the following primary commands on this panel:

SAVE Saves the specified product or component parameter values.

VERIFY / VERIFYOFF

Use the VERIFY and VERIFYOFF commands to turn on and off parameter verification of DB2 parameters. Before you can generate customization jobs, you must verify that all required parameters are set to a valid value. The DB2 parameter status of Verify Values on the Customize Workplace panel indicates that the values have not been verified.

Enter these commands either by typing them in the command field and pressing Enter or by positioning the cursor on the command and pressing Enter. When VERIFY is active, VERIFYOFF is displayed so that you can toggle between the two states. By default, verification is turned on when you display the DB2 Parameters panel, and the verification state is reset to VERIFY every time you exit the Customization Workplace panel by pressing PF3.

Turning verification off is useful when you need to exit the panel before you have entered all of the required parameters, but you want to save the parameters that you have specified. When you disable verification, it is disabled only for the DB2 Parameters Values 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, Verify Values, or Discovered, edit these parameters.

Related tasks:

“Defining DB2 HPU parameters” on page 39

DB2 HPU parameters are specific to DB2 HPU.

“Defining LPAR parameters” on page 41

LPAR parameters are parameters on the local LPAR that are required to customize DB2 HPU.

Generating customization jobs

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

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 HPU parameters, LPAR 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 HPU.

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.

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 : XYZ.HINZ430.SINZDENU > LPAR . . . : ZPS1
 Product name : IBM DB2 High Performance U > Version . . : 4.3.0

Line Commands: E - Edit B - Browse

Product customization library . : XYZ.HINZ430.\$ZPS1\$.INZ430 >

Cmd	Member	SSID	GrpAttch	Template	Date	Description
-	A0TVAR	--	--	INZTVAR	2012/06/06	Build the member of parameters
	A1PARM	--	--	INZPARM	2012/06/06	Define DSNAME of PARMLIB to loa
	A2HPUCL	--	--	INZHPUCL	2012/06/06	Generate CLIST to start HPU
	A3LAUNC	--	--	INZLAUNC	2012/06/06	Integrate DB2 HPU to DB2 Tools
	A4ADT00	--	--	INZADT00	2012/06/06	Integrate DB2 HPU into DB2 Admi
	A5BIAAAA	--	DG93	INZBIND	2012/03/19	BIND and GRANT for HPU
	A5BIAAAD	--	DB0A	INZBIND	2012/03/19	BIND and GRANT for HPU
	A5BIAAAG	DZ7M	--	INZBIND	2012/03/19	BIND and GRANT for HPU
	A5BIAAAH	DZ8A	--	INZBIND	2012/03/19	BIND and GRANT for HPU
	A5BIAAAI	DZ8C	--	INZBIND	2012/03/19	BIND and GRANT for HPU
	A5BIAAAJ	DZ8D	--	INZBIND	2012/03/19	BIND and GRANT for HPU
	A5BIAAAM	DZ8E	--	INZBIND	2012/03/19	BIND and GRANT for HPU
	A5BIAAAN	DZ9E	--	INZBIND	2012/03/19	BIND and GRANT for HPU
	A5BIAAAO	DZAA	--	INZBIND	2012/03/19	BIND and GRANT for HPU
	A5BIAAAR	DZAJ	--	INZBIND	2012/03/19	BIND and GRANT for HPU
	A5BIAAAS	DG93	--	INZBIND	2012/03/19	BIND and GRANT for HPU
	A5BIAAAT	DZ8B	--	INZBIND	2012/03/19	BIND and GRANT for HPU
	A5BIAAAV	DB0A	--	INZBIND	2012/03/19	BIND and GRANT for HPU
	A5BIAAAW	DZAE	--	INZBIND	2012/03/19	BIND and GRANT for HPU
	A5BINDAA	--	DG93	INZBIND	2012/06/06	BIND and GRANT for HPU
	A5BINDAD	--	DB0A	INZBIND	2012/06/06	BIND and GRANT for HPU
	A5BINDAG	DZ7M	--	INZBIND	2012/06/06	BIND and GRANT for HPU

Figure 12. The Finish Product Customization panel

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

The New filed indicates if the job member is newly created/updated. It is either YES or NO. YES indicates the job member is newly created or updated, and it needs to be submitted for customization. NO indicates the job member is not newly created/updated, it does not need to be submitted for customization.

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 HPU assigns the template name.

DB2_entry_ID

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

For example, the XYZBNDDDB2_entry_ID_1 and XYZBNDDDB2_entry_ID_2 jobs are generated from the XYZBNDGR template, and the XYZ4DB2_entry_ID_1 and XYZ4DB2_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 an LPAR or 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 HPU 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 or E against a customization job or the product customization library, and press Enter. An ISPF browsing or editing 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 HPU is customized, and the Customizer Workplace panel is displayed. The status is Customized for the DB2 entries on which DB2 HPU 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 HPU.

Browsing parameters

You can browse the product or component parameters, the LPAR parameters, and the DB2 parameters in read-only mode.

Procedure

1. On the Customizer Workplace panel, specify B next to the **Product parameters** field, the **LPAR 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 15

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 15

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 HPU 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 HPU 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 HPU 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.

Optional: Integrating DB2 HPU into DB2 Tools Launchpad

Optionally, you can integrate DB2 HPU into DB2 Tools Launchpad. Tools Customizer will create the necessary JCL, but you must manually complete some steps after you submit the customization job.

Before you begin

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

- The DB2 Tools Launchpad is installed.
- Tools Customizer generated the job from the INZLAUNC template, and you submitted the job.

About this task

The DB2 Tools Launchpad is a centralized panel from which you can launch integrated DB2 Tools. After you integrate DB2 HPU, you can launch DB2 HPU from the DB2 Tools Launchpad.

Procedure

1. Run the CLIST INZADBI in SINZCLST. The CLIST uses the high-level qualifier that you specify for the DB2 Admin data sets and the name of the library that contains the ADBDMTI EXEC. The DB2 Tools Table – ADD An Entry panel is displayed, as shown in the following figure:

```

----- DB2 Tools Table - ADD An Entry ----- 08:57
Command ==>

Tool Name : HPU High Performance Unload
Code      : HPU      (User-defined code, for shortcut tool identifier)
Prog No.  : 5655-AA1 (IBM program product number or equivalent)
Release   : 420      (Product release number)
Group     : 2        (Tool category, as follows:
                    1 - Administration Tools
                    2 - Application Management Tools
                    3 - Performance Management Tools
                    4 - Recovery and Replication Management)
Installed : Y        (Y - yes, N - no)
Command   : SELECT CMD(INZHPU D LP )

```

Figure 13. Adding an entry to the DB2 Tools Launchpad panel

2. Press Enter to confirm the new DB2 HPU command.

Results

When INZADBI completes successfully, a new line, HPU, is added to the DB2 Tools Launchpad.

Optional: Integrating DB2 HPU into DB2 Administration Tool

Optionally, you can integrate DB2 HPU into DB2 Administration Tool (DB2 Admin). Tools Customizer will create the necessary JCL, but you must manually complete some steps after you submit the customization job.

Before you begin

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

- DB2 Admin is installed.
- Tools Customizer generated the job from the INZADTOO template, and you submitted the job.

About this task

DB2 Admin helps you manage DB2 environments efficiently and effectively. After you have integrated DB2 HPU, you can run DB2 HPU unload operations by using DB2 Admin.

Procedure

1. Follow the instructions that are described in INZDB21X in the SINZCLIST library. You can customize these instructions by changing the name of the library which contains DB2 Admin commands tables and the name of the library which contains the ADBDMTI EXEC.

Requirement: Before you can use the INZDB21X member, you must generate it with Tools Customizer whenever you change the library names.

2. Run the ADB21S and ADB21T procedures that you have modified. These procedures are used to re-create the DB2 Admin Tool command tables.

Results

When ADB21S and ADB21T complete successfully, you can start DB2 HPU interactively by using the DB2 Admin.

Related reference:

“INZDB21X member” on page 456

Use the INZDB21X member to integrate DB2 HPU with DB2 Administration Tool.

Chapter 3. Input and output

DB2 HPU uses various sources of input data, various formats for output data, and a user exit to customize the output data set that is created by a SELECT statement.

Topics:

- “Input data sources”
- “Output formats” on page 61
- “Specifying output file options” on page 65
- “LOB data processing” on page 68
- “XML data processing” on page 68
- “Output data consistency” on page 69

Input data sources

DB2 HPU can unload from various input data sources.

Topics:

- “Nonpartitioned table spaces”
- “Partitioned table spaces”
- “Full image copies or incremental image copies” on page 58
- “Last full image copies” on page 60

Nonpartitioned table spaces

DB2 HPU can use nonpartitioned table spaces as input data.

If the selected tables belong to the same table space, multiple unload requests can be processed on this object in parallel.

Example: Unloading tables in parallel

The *mydb.myTS01* table space contains the *me.myTable01* table and the *me.myTable02* table. The UNLOAD command unloads both tables in parallel, as shown in the following example:

```
UNLOAD TABLESPACE mydb.myTS01

      SELECT * FROM me.myTable01
      OUTDDN (DDNTBL01)
      FORMAT VARIABLE END

      SELECT * FROM me.myTable02
      OUTDDN (DDNTBL02)
      FORMAT DSNTIAUL
```

Partitioned table spaces

DB2 HPU can use partitioned table spaces as input data.

DB2 HPU can use as input all partitions or a subset of partitions. As with nonpartitioned table spaces, DB2 HPU can process multiple unload requests in parallel.

The following examples show how to unload data in partitioned table spaces. In these examples, the *mydb.myTS02* table space has five partitions.

Example: Unloading a partitioned table space

This example shows how to unload all partitions in the *mydb.myTS02* table space. In this example, a single OUTDDN ddname, MYOUT, is allocated in the JCL. All rows are written in this ddname. If a ddname per partition is allocated in the JCL, such as MYOUT01, MYOUT02...MYOUT05, the rows from each partition are unloaded in the corresponding ddname.

```
UNLOAD TABLESPACE mydb.myTS02
SELECT * FROM me.mytable01
OUTDDN(MYOUT)
FORMAT DSNTIAUL
```

Example: Unloading specific partitions

This example shows how to unload partitions 1, 3, and 4 only.

```
UNLOAD TABLESPACE mydb.myTS02 PART (1,3,4)
SELECT * FROM me.myTable01
OUTDDN (MYOUT)
FORMAT VARIABLE END
```

Example: Unloading partition subsets by using multiple SELECT statements

This example shows how to unload different subsets of partitions from a single table space by using multiple SELECT statements. Instead of specifying the list of partitions to be unloaded at the table space level, you can specify the partitions at the SELECT level.

```
UNLOAD TABLESPACE mydb.myTS02
SELECT * FROM me.myTable01 PART (1,3,4)
OUTDDN (MYOUT)
FORMAT VARIABLE END
SELECT * FROM me.myTable01 PART (2,5)
OUTDDN (MYOUT1)
FORMAT DSNTIAUL
```

Example: Unloading a partitioned table space with one output file per partition in DB2 FORCE when an unsupported SELECT statement is used

This example shows how to unload data from partitions to get one output file per partition when an unsupported SELECT statement is used. The SELECT statement is not supported because the WHERE clause uses a subselect. A separate output file per partition is allowed because SQLPART (ALL) and a template that contains the &PART variable are specified.

```
TEMPLATE OUTFILE DSN qua11.&DB..&TS..P&PART. UNIT SYSDA
SPACE(200, 200) CYL
UNLOAD TABLESPACE mydb.myTS02 DB2 FORCE
SELECT * FROM me.myTable01 SQLPART (ALL)
WHERE COL1 IN (SELECT COLREF FROM Reference.Table)
OUTDDN (OUTFILE)
FORMAT DSNTIAUL
```

Related concepts:

“Full image copies or incremental image copies”

DB2 HPU can use full image copies (FIC) or incremental image copies as input data.

Full image copies or incremental image copies

DB2 HPU can use full image copies (FIC) or incremental image copies as input data.

When DB2 HPU unloads data from an image copy, it can process only physical unloads or logical unloads that correspond to SELECT statements that are natively supported.

When you specify the COPYDDN keyword, DB2 HPU can obtain input data from an FIC or an incremental image copy. The rules for processing are the same as the rules that apply to a table space:

- If the image copy is of the entire table space, DB2 HPU processes every partition by default unless you have requested partition filtering by using the PART keyword at the UNLOAD level or at the SELECT level.
- If the image copy is of a single partition of the table space, DB2 HPU processes the specific partition that is contained in the image copy.

If the input image copy is a global copy of an entire partitioned table space, DB2 HPU can generate one output file for every partition by allocating a ddname per unloaded partition in the JCL, or it can generate one output file per partition by using a TEMPLATE statement that contains the &PART variable.

When the input FIC corresponds to a table space that contains a single table, the header page contains the object ID (OBID) for this table. When this input FIC is unloaded and the ORIGINOBID keyword is specified, DB2 HPU checks to see whether both OBIDs match. If both OBIDs do not match, the unload is not processed, and an error message is issued.

The ddname that is specified in the COPYDDN keyword can correspond to a JCL-allocated file or to a TEMPLATE definition.

The following examples show how DB2 HPU uses FICs and incremental image copies as input to generate output files:

Example: Unloading data from an image copy of an entire partitioned table space in a single output file

The *mydb.myTS02* table space has five partitions. The COPYDDN ddname points to an image copy of this table space.

```
UNLOAD TABLESPACE mydb.myTS02
COPYDDN COPYDDN
SELECT * FROM me.myTable01
OUTDDN (MYOUT)
FORMAT VARIABLE END
```

Example: Unloading data from an image copy of an entire partitioned table space with partition filtering and with one output file per partition

The *mydb.myTS02* table space has five partitions. The COPYDDN ddname corresponds to a JCL-allocated file and points to a full image copy of this table space. One output file per partition is requested by using the &PART variable in the TEMPLATE definition for the output file. Partition filtering is requested by specifying the PART keyword in the UNLOAD command.

```
TEMPLATE OUTFILE DSN &USERID..&DB..&TS..U.P&PART. UNIT SYSDA
UNLOAD TABLESPACE mydb.myTS02 PART(1:3)
COPYDDN COPYDDN
SELECT * FROM me.myTable01
OUTDDN (OUTFILE)
FORMAT DSNTIAUL
```

Example: Unloading data from an image copy with COPYDDN and DDLDDN as input files

The COPYDDN1 and DDLDDN1 ddnames are used as input files.

```

//UNLOAD EXEC PGM=INZUTILB,PARM='DB81,DB2UNLOAD',
// REGION=0M,TIME=1440
//STEPLIB DD DSN=DB2UNL.SINZLINK,DISP=SHR
// DD DSN=DSN810.SDSNLOAD,DISP=SHR
//CPYDDN1 DD DISP=SHR,DSN=HLQ.FIC
//SYSREC DD DISP=(NEW,CATLG,DELETE),DSN=HLQ.UNLOAD,
// SPACE=(CYL,(5,5),RLSE),UNIT=SYSDA
//DDLDDN1 DD DISP=SHR,DSN=HLQ.DDL
//SYSIN DD *
UNLOAD TABLESPACE
COPYDDN CPYDDN1 DDLDDN DDLDDN1
SELECT *
FROM me.myTable01
ORIGINOBID 0
OUTDDN (SYSREC)
FORMAT DSNTIAUL

```

Related tasks:

“Specifying one output file per partition” on page 66

You can unload each partition in a table space to a separate file.

Last full image copies

DB2 HPU can use last full image copies as input data.

You can request DB2 HPU to unload data from the last full image copy that was taken for a table space by specifying COPYDDN LAST_IC, as shown in the following example:

```

UNLOAD TABLESPACE mydb.myTS02
COPYDDN LAST_IC
SELECT * FROM me.myTable01
OUTDDN (MYOUT)
FORMAT VARIABLE END

```

The last full image copy of a partitioned table space can be an image copy of the entire table space or an image copy of each partition. You can control which types of image copies are used by specifying various keywords.

- By using the GLOBAL keyword, you can specify DB2 HPU to use only global image copies.
- By using the PARTITIONED keyword, you can specify DB2 HPU to use only image copies of partitioned table spaces.
- By using the ANYTYPE keyword, you can specify DB2 HPU to use any type of image copy.
- By using the CONSISTENT keyword, you can ensure that the image copies of all unloaded partitions have the same START_RBA value in the SYSCOPY column.

The following example shows how you can unload data from a consistent image copy per partition with one output file per partition.

Example: Unloading data from a consistent image copy per partition of a partitioned table space with one output file per partition

The *mydb.myTS02* table space has five partitions. One image copy per partition has been taken in a single COPY invocation. The ANYTYPE keyword authorizes DB2 HPU to use these image copies, if they are the most recent for each partitions. The CONSISTENT keyword ensures that all the image copies that are used as input have the same START_RBA value in the SYSCOPY column. One output file per partition is requested by using the &PART variable in the TEMPLATE definition for the output file.

```
TEMPLATE OUTFILE DSN &USERID..&DB..&TS..U.P&PART. UNIT SYSDA
UNLOAD TABLESPACE mydb.myTS02
COPYDDN LAST_IC ANYTYPE CONSISTENT
SELECT * FROM me.myTable01
OUTDDN (OUTFILE)
FORMAT DSNTIAUL
```

Related reference:

“UNLOAD block syntax and description” on page 110

Use the UNLOAD statement to specify the data to be unloaded from a table space or an image copy. The UNLOAD block contains the OPTIONS and SELECT blocks.

Output formats

DB2 HPU can create output files in various formats for physical and logical unloads.

Use a physical unload to unload the entire table space, which includes all tables, columns, and rows unless sampling was requested. Use a logical unload to filter the rows, select the columns, and compute the output fields that you want to unload by using SELECT statements.

Topics:

- “Performing physical unloads”
- “Performing logical unloads” on page 62
- “Output encoding scheme” on page 65

Performing physical unloads

A physical unload starts unloading at the beginning of a table space and does not stop unloading until the entire table space is unloaded unless sampling is requested. A physical unload uses the same format as the IBM DB2 REORG utility to physically unload the table space.

About this task

The main use of data that is unloaded by a physical unload is to reload it in a table space that has the same structure or the same table space by using a LOAD with FORMAT UNLOAD.

When you use the REORG unload-only format, the table space is unloaded regardless of the tables that it contains.

You must use an exit to select the rows to unload or to change the format of these rows. You can limit the number of rows to be unloaded and take a sample of some rows. The sampling is done before the call to the exit.

Procedure

Specify either the table space name or a LISTDEF statement in the UNLOAD command, and use the UNLDDN parameter in the UNLOAD command.

Example

The following example shows how you can take a sample of some rows when you do a physical unload.

Example: Sampling rows to be unloaded

The *mydb.myTS01* table space contains two tables. One out of every 10 rows is unloaded until the maximum number of rows is reached. The maximum number of rows is 150.

```
UNLOAD TABLESPACE mydb.myTS01
UNLDDN MYDDN
UNLMAXROWS 150
UNLFREQROWS 10
```

Important: When you do a physical unload, you can code one or more SELECT statements. However, you can have only one UNLDDN keyword per UNLOAD command.

The following example shows how you can unload data by using LISTDEF and TEMPLATE statements.

Example: Unloading data by using LISTDEF and TEMPLATE statements

The *mydb* database contains three table spaces. The LISTDEF and TEMPLATE statements unload each table space in a separate file with a DSN that includes the complete table space name and the date.

```
LISTDEF LIST1 INCLUDE TABLESPACE mydb.myTS%
TEMPLATE DDUNL DSN HLQ.&DB..&TS..D&DATE..UNLOAD
UNLOAD TABLESPACE LIST(LIST1)
UNLDDN DDUNL
```

Performing logical unloads

A logical unload uses SELECT statements to filter the rows and columns that you want to unload. Use a logical unload to determine the exact content of the output data by using an SQL SELECT statement and to specify the output format.

Procedure

1. Specify one or more SELECT statements. The SELECT statement can specify the name of a table or view, or you can use a LISTDEF statement to automatically generate a logical unload for each table from each table space in the LISTDEF. You can use SQL comment indicators in these SELECT statements.
2. Specify the OUTDDN keyword for each SELECT statement to specify the output data set for each SELECT statement. The OUTDDN keyword can refer to a JCL-allocated file or to a template to use dynamic data set allocation.
3. Select one of the following output formats: DSNTIAUL, DELIMITED, VARIABLE, USER, EXTERNAL, or INTERNAL. The following example shows how you can unload data by using SELECT statements.

Example: Unloading data by using SELECT statements

The *mydb.myTS01* table space contains two tables. One out of every 10 rows is unloaded until the maximum number of rows is reached. The maximum number of rows is 150. The SELECT statements request a logical unload of the same two tables.

```
UNLOAD TABLESPACE mydb.myTS01
UNLDDN MYDDN
UNLMAXROWS 150
UNLFREQROWS 10
  SELECT * FROM me.myTable01
  OUTDDN (MYOUTTB1)
  FORMAT VARIABLE END
  SELECT * FROM me.myTable02
  OUTDDN (MYOUTTB2)
  FORMAT VARIABLE END
```

Details about the various output formats are provided in the following sections.

DSNTIAUL format

When you create output in the DSNTIAUL format, the output is identical to the output that is produced by the DSNTIAUL program.

The following example shows how to unload a table in DSNTIAUL format.

Example: Unloading a table in DSNTIAUL format

In this example, the *mydb.myTS01* table space contains two tables. This example shows how to unload the *myTable01* table in DSNTIAUL format.

```
UNLOAD TABLESPACE mydb.myTS01
SELECT * FROM myQual.myTable01
OUTDDN (MYDDN)
FORMAT DSNTIAUL
```

DELIMITED format

When you create output in the DELIMITED format, you can specify a separator character and a delimiter character.

CHAR, VARCHAR, GRAPHIC, and VARGRAPHIC columns are enclosed by the delimiter character. Null columns are not enclosed by the delimiter character if DELIM *val* and NULL DELIM are coded.

The following example shows how to use a delimiter and a separator when you unload a table.

Example: Unloading a table by using a delimiter and a separator

In this example, the *mydb.myTS01* table space contains two tables. The example shows how to unload a table by using an asterisk (*) as the delimiter and a semicolon (;) as the separator.

```
UNLOAD TABLESPACE mydb.myTS01
SELECT * FROM myQual.myTable01
OUTDDN (MYDDN)
FORMAT DELIMITED SEP ';' DELIM '*'
```

Related reference:

“DB2 HPU output data parameters” on page 410

The DB2 HPU output data parameters section on the Product Parameters panel (CCQPPRD) in Tools Customizer contains the parameters for configuring output data.

VARIABLE format

When you create output in the VARIABLE format, the output is compatible with the DB2 LOAD utility input data set.

The following example shows how to unload a table so that all variable-length fields are treated as variables.

Example: Unloading a table so that all variable-length fields are treated as variables

In this example, table space *mydb.myTS01* contains two tables. The example shows how to unload a table so that all the variable-length fields are treated as variables.

```
UNLOAD TABLESPACE mydb.myTS01
SELECT * FROM myQual.myTable01
OUTDDN (MYDDN)
FORMAT VARIABLE ALL
```

Related reference:

“DB2 HPU output data parameters” on page 410

The DB2 HPU output data parameters section on the Product Parameters panel (CCQPPRD) in Tools Customizer contains the parameters for configuring output data.

USER format

When you create output in the USER format, you can customize every output column in any manner that you want.

For example, you can force the conversion between data types, change the date or time format, add or remove a length field, add or remove a null indicator, justify the content left or right, select a padding character, select a delimiter character for date or time, and so on.

The following example shows how to change columns:

Example: Changing columns

In this example, the *mydb.myTS01* table space contains two tables. The example shows how to change column 1 from a VARCHAR(10) to a CHAR(15) with null indicator, pad the column with the number sign (#) and left align it. The other columns are unloaded in the default format.

```
UNLOAD TABLESPACE mydb.myTS01
SELECT * FROM myQual.myTable01
OUTDDN (MYDDN)
FORMAT USER (
  COL 1 TYPE CHAR(15)
        PADDING '#'
        NULLID YES
        JUST LEFT
)
```

The following example shows how to concatenate columns.

Example: Concatenating columns

In this example, the *mydb.myTS01* table space contains two tables. The example shows how to concatenate columns FIRSTNAME, LASTNAME, and COMPANY to create a list of e-mail addresses. This example assumes that all these columns are VARCHAR.

```
UNLOAD TABLESPACE mydb.myTS01
SELECT FIRSTNAME, '.', LASTNAME, '@',
COMPANY, '.com' FROM myQual.myTable01
OUTDDN (MYDDN)
FORMAT USER (
  COL FIRSTNAME
        NULLID NO
        LENGTH REAL
        LENGTHBYTE NO ,
  COL LASTNAME
        NULLID NO
        LENGTH REAL
        LENGTHBYTE NO ,
  COL COMPANY
        NULLID NO
        LENGTH REAL
        LENGTHBYTE NO
)
```

The following output is the result:

```
john.smith@company.com
```


EXTERNAL format

When you create output in the EXTERNAL format, output fields are in the EXTERNAL format that corresponds to their default type, output records are fixed, and a field separator is not used.

The following example shows how to unload each file in the EXTERNAL format and sort the records in the table clustering index order.

Example: Unloading files in FORMAT EXTERNAL and sorting the records in the table clustering index order

In this example, the *mydb* database contains three table spaces: *myTS01*, *myTS02*, and *myTS03*. Each of these table spaces contains three tables: *me.myTable01A*, *me.myTable01B*, and *me.myTable01C* for *myTS01*; *me.myTable02A*, *me.myTable02B*, and *me.myTable02C* for *myTS02*; and *me.myTable03A*, *me.myTable03B*, and *me.myTable03C* for *myTS03*.

```
LISTDEF LIST1 INCLUDE TABLESPACE mydb.myTS%
TEMPLATE DDUNL DSN HLQ.&DB..&TS..D&DATE..T&SEL.
GLOBAL OPTIONS TEMPLATESET( SEL = :SELNUM ) ;
UNLOAD TABLESPACE
SELECT * FROM LIST(LIST1) ORDER CLUSTER OUTDDN DDUNL FORMAT EXTERNAL
```

Related reference:

“DB2 HPU output data parameters” on page 410

The DB2 HPU output data parameters section on the Product Parameters panel (CCQPPRD) in Tools Customizer contains the parameters for configuring output data.

Output encoding scheme

For the DSNTIAUL, DELIMITED, VARIABLE, USER, and EXTERNAL output formats, the translations from EBCDIC to ASCII and from ASCII to EBCDIC are supported only for single-byte character set (SBCS) character strings. Data is translated by using the translation tables in the SYSIBM.SYSSTRINGS table.

Other types of translation are done by using Unicode Conversion Services.

Related reference:

“OPTIONS block syntax and description” on page 134

Use the OPTIONS block to specify the default conversions that are with the SELECT statements. This block can be used in the GLOBAL block, the UNLOAD block, and the SELECT block.

Specifying output file options

You can override the automatic output file allocation, specify one output file per partition, or limit the number of unloaded rows and call a user exit.

Topics:

- “Overriding the automatic output file allocation”
- “Specifying one output file per partition” on page 66
- “Limiting the number of unloaded rows and calling a user exit” on page 67

Overriding the automatic output file allocation

DB2 HPU automatically calculates the output file allocation by using the DCB parameters that you provided in the JCL for each OUTDDN and UNLDDN that is specified in the SYSIN. However, you can override the automatic allocation.

Before you begin

Ensure that the allocation that you want to specify is compatible with the unload that you request by ensuring that the record length is long enough to contain the longest row that you want to unload. If you do not specify a DCB parameter in the DD statement, DB2 HPU uses the calculated DCB.

About this task

To override the automatic output file allocation:

Procedure

1. Ensure that the syntax that is used allows you to override the output file allocation, and specify DFSIGDCB YES.
2. Code DD statements in the JCL, or specify a TEMPLATE definition. DB2 HPU generates the output data sets based on the JCL or the TEMPLATE definition.

Specifying one output file per partition

You can unload each partition in a table space to a separate file.

Procedure

To specify one output file per partition, do one of the following steps:

- Use a TEMPLATE with a dsname that contains the &PART variable or code a base ddname in the OUTDDN statement, and declare this ddname in your JCL.
- Use the partition number as a suffix on the base ddname.

The following example shows how to specify one output file per iteration by using the partition number as a suffix.

Example: Specifying one output file per iteration by using the partition number as a suffix on the base ddname

In this following example, the *mydb.myTS02* table space is a partitioned table space with five partitions. The example shows how to unload only partitions 1, 3, and 4 into separate output files by using the partition number as a suffix on the base ddname. The base ddname is MYOUT.

```
UNLOAD TABLESPACE mydb.myTS02 PART (1,3,4)
```

```
SELECT * FROM me.myTable01  
OUTDDN (MYOUT)  
FORMAT VARIABLE END
```

In the JCL, specify the following DD statements:

```
MYOUT001 DD ...  
MYOUT003 DD ...  
MYOUT004 DD ...
```

The following example shows how to specify one output file per iteration by using a TEMPLATE statement.

Example: Specifying one output file per iteration by using a TEMPLATE statement

In this example, the *me.myTable02* table was created in a partitioned table space with 128 partitions. The example shows how to unload all partitions

into separate output files. The unloaded data is sorted in the order of the clustering index, and the output files are dynamically allocated by using a template named PARTDS.

```
TEMPLATE PARTDS DSN HLQ.&DB..&TS..D&DATE..P&PART.  
UNLOAD TABLESPACE  
SELECT * FROM me.myTable02  
ORDER CLUSTER  
OUTDDN PARTDS FORMAT DSNTIAUL
```

The following example shows how to specify one output file per partition in an unload from a global full image copy.

Example: Specifying one output file per partition in an unload from a global full image copy

In this example, a global FIC exists for the DBINFDM.TSPART6 partitioned table space. This FIC is allocated in the JCL by using the COPYDD statement.

```
TEMPLATE SYSREC DSN IBMUSER.&DB..&TS..P&PART.  
UNLOAD TABLESPACE DBINFDM.TSPART6  
COPYDDN COPYDD  
SELECT * FROM TBPART6  
OUTDDN SYSREC  
FORMAT DSNTIAUL
```

Limiting the number of unloaded rows and calling a user exit

An OUTDDN statement can be followed by one of several keywords that you can use to limit the number of unloaded rows, sample these rows, and call a user exit.

Procedure

1. To limit the number of rows, do one of the following steps:
 - For a physical unload, use the UNLMAXROWS and UNLFREQROWS keywords.
 - For a logical unload, use the OUTMAXROWS and OUTFREQROWS keywords.
 - For a SELECT statement that is processed by DB2, the OUTMAXROWS and OUTFREQROWS keywords are always applied on rows that are returned by DB2 after the WHERE and ORDER BY clauses are applied.
 - By default, for a SELECT statement that is natively processed by DB2 HPU, the OUTMAXROWS and OUTFREQROWS are applied when the rows are read in the VSAM LDS files. To be consistent with a SELECT statement that is processed by DB2, use the optional keyword ON_RESULT_TABLE after OUTMAXROWS and OUTFREQROWS.
2. To request that each output row that is produced by DB2 HPU be processed by a user exit before it is written to the output file, use the OUTEXIT option.

Example

The following example shows how to unload partitions into separate output files, sample rows, and call a user exit.

Example: Unloading partitions into separate output files, sampling rows, and calling a user exit

In this example, the *mydb.myTS02* partitioned table space has five partitions. The example shows how to unload only partitions 1, 3, and 4 into separate output files. One row out of 10 is selected and passed to a

COBOL/2 user exit that applies user modifications before output. The unload process stops when 150 rows have been selected.

Requirement: The exit must be in STEPLIB or in LINKLIST and must reside in an authorized library.

```
UNLOAD TABLESPACE mydb.myTS02 PART (1,3,4)
```

```
SELECT * FROM me.myTable01
OUTDDN (MYOUT) OUTMAXROWS 150
        OUTFREQROWS 10
        OUTEXIT myExit COBOL2
FORMAT VARIABLE END
```

LOB data processing

DB2 HPU can unload LOB data.

Supported LOB data sources

You can unload LOB input data on the base table space, but support is limited by the following restrictions.

- DB2 HPU cannot directly unload a LOB table space.
- DB2 HPU does not support the unloading of LOB data from non-flash image copies.
- Unloading LOB data is only supported for a logical unload. When a physical unload is done on a base table space that contains a table with an LOB column, the LOB data is not unloaded.

LOB output

Unloading LOB data in the same output file that contains the remaining columns of the table is called *in-stream processing*. In-stream processing is supported only when the SELECT statement is processed by DB2, which means that DB2 FORCE or DB2 YES was specified along with an unsupported SELECT statement. When this method is used, DB2 HPU might have to truncate the LOB data to limit the record size to the maximum size that is allowed. Unloading LOB data by using a LOB file reference is supported in all cases (DB2 NO, DB2 YES, or DB2 FORCE), and truncation cannot occur. To use a LOB file reference, specify the BLOBF, CLOBF, or DBCLOBF option in a REFORMAT clause, an INTO clause, or a USER format definition.

Related reference:

“Example: Unloading data from a table with a LOB column by using a TEMPLATE statement” on page 104

This example shows how to use a TEMPLATE statement to unload data from a table with a LOB column (COL_LOB).

XML data processing

DB2 HPU can unload XML data.

Supported XML data sources

You can unload XML input data from the base table space, but support is limited by several restrictions.

- DB2 HPU cannot directly unload an XML table space.

- DB2 HPU does not support the unloading of XML data from non-flash image copies.
- Unloading XML data is supported only for a logical unload. When a physical unload is done on a base table space that contains a table with an XML column, the XML data is not unloaded.

XML output

Unloading XML data in the same output file that contains the remaining columns of the table is called *in-stream processing*. It is supported only when the SELECT statement is processed by DB2, but truncation of the XML data is not supported. Processed by DB2 means that DB2 FORCE or DB2 YES was specified along with an unsupported SELECT statement. When this method is used, if an XML value is longer than the output field, an SQL CODE -433 error is issued by DB2, and processing ends.

Unloading XML data by using a LOB file reference is supported when DB2 NO, DB2 YES, or DB2 FORCE is specified. Truncation cannot occur. To use a LOB file reference, specify the BLOBF, CLOBF, or DBCLOBF option in a REFORMAT clause, an INTO clause, or a USER format definition.

Related reference:

“Example: Unloading data from a table with an XML column by using a TEMPLATE statement” on page 103

This example shows how to use a TEMPLATE statement to unload data from a table with an XML column.

Output data consistency

Ensuring output data consistency depends on the data source and the settings of various options.

The data consistency of the output during the unload process depends on the following data sources and keywords:

Data sources

- A table accessed through DB2 by SQL access
- The table space that contains the table by direct access
- An image copy data set of the table space

Keywords

- SQLACCES and QUIESCECAT
- LOCK and QUIESCE

Topics:

- “Interaction with the DB2 catalog”
- “Consistency considerations” on page 71

Interaction with the DB2 catalog

DB2 HPU can use direct access, SQL access, or both to read the descriptions of the unloaded objects in the DB2 catalog. Output data consistency is affected by the access method that you specify.

Direct access

Direct access provides direct DB2 HPU access to the VSAM files that

contain the DB2 catalog table spaces. Direct access is generally faster because it uses the internal pointers found in the DB2 catalog between unloaded objects such as table spaces, partitions, tables, indexes, and columns).

If the content of the catalog has recently changed because of a DROP, CREATE, or ALTER command, or because a utility has updated the catalog, issue a QUIESCE to ensure that the VSAM files have the most current information. For example, you might want to issue a QUIESCE when an online reorganization changes the value of the IPREFIX column in the SYSTABLEPART table. The QUIESCE ensures that DB2 HPU can read accurate data from the VSAM file because the content of the DB2 buffers is externalized on the DASD. A high degree of output data consistency is ensured.

Depending on the security product that is used at your site, you might need READ access to the VSAM LDS files that contain the DB2 catalog when direct access is used.

SQL access

DB2 HPU can use SQL to access the DB2 catalog. This method returns the most current value of information that is found in the DB2 catalog. No QUIESCE of the DB2 catalog is required. Although a high degree of output data consistency is attained, SQL access can lead to a table space scan on the table spaces of the DB2 catalog, mainly on DSNDB06.SYSDBASE. Performance might be affected because no index exists for some of the access made by DB2 HPU, such as when the list of tables in the unloaded table space is retrieved.

The VUM027/SQLACCES parameter is used by DB2 HPU to determine which method to use when accessing the DB2 catalog. When VUM027/SQLACCES is set to YES or MINIMAL, or when VUM027/SQLACCES is set to AUTO and QUIESCECAT is not requested, the DB2 catalog is accessed by using dynamic SQL. In this case, the user ID that is used to access the DB2 catalog depends on the setting of the VUM032/ACTLGUSR parameter. The following table shows which tables of the DB2 catalog require SELECT authority for the user ID:

Table 11. Tables of the DB2 catalog that require SELECT authority

Settings of VUM027/SQLACCES	Tables
MINIMAL	- SYSIBM.SYSTABLEPART - SYSIBM.SYSINDEXPART - SYSIBM.SYSTABLESPACE - SYSIBM.SYSTABLES

Table 11. Tables of the DB2 catalog that require *SELECT* authority (continued)

Settings of VUM027/SQLACCES	Tables
<ul style="list-style-type: none"> • AUTO without QUIESCECAT • YES 	<ul style="list-style-type: none"> - SYSIBM.SYSTABLEPART - SYSIBM.SYSINDEXPART - SYSIBM.SYSTABLESPACE - SYSIBM.SYSTABLES - SYSIBM.SYSCOLUMNS - SYSIBM.SYSINDEXES - SYSIBM.SYSKEYS - SYSIBM.SYSRELS - SYSIBM.SYSFOREIGNKEYS - SYSIBM.SYSVTREE - SYSIBM.SYSVIEWS - SYSIBM.SYSVIEWDEP - SYSIBM.SYSFIELDS
<ul style="list-style-type: none"> • MINIMAL • AUTO • YES • NO 	<p>Important: The following tables are accessed by using dynamic SQL only when a LISTDEF is used in the SYSIN.</p> <ul style="list-style-type: none"> - SYSIBM.SYSTABLES - SYSIBM.SYSTABLESPACE - SYSIBM.SYSINDEXES - SYSIBM.SYSRELS

The SYSIBM.SYSCOPY table is always accessed by using SQL when you unload data from an image copy.

Related reference:

“DB2 HPU DB2 parameters” on page 397

The DB2 parameters section on the Product Parameters panel (CCQPPRD) in Tools Customizer contains the DB2 parameters that are used by DB2 HPU.

Consistency considerations

You should understand the consistency considerations when you unload data from an image copy or from a table.

Data that is unloaded from an image copy

When you unload data from an image copy of the table space, the level of consistency is inherited from the SHRLEVEL option that was run with the COPY utility.

When you unload data from an image copy, the updates on the table at run time have no impact on the output data. Therefore, do not use the QUIESCE and LOCK keywords in this case because they are ignored.

When you unload data from image copies that are taken per partition with the COPYDDN LAST_IC PARTITIONED, COPYDDN LAST_IC ANYTYPE, or COPYDDN *number* PARTITIONED keywords, you can ensure the consistency between the image copies of each partition by using the CONSISTENT keyword so that all input full image copies that were used have been taken at the same START_RBA.

Data that is unloaded from a table or its underlying table space

When you unload from the table itself (the online data that is handled by DB2), the DB2, QUIESCE, and LOCK keywords can affect the output data consistency.

DB2 DB2 HPU retrieves the data from the table by using one of the following methods:

Direct access

When the SELECT statement can be evaluated by DB2 HPU, the data is read directly from the table space linear data sets (LDS) that contain the table and the related index space LDS, when they are applicable.

SQL access

When the SELECT statement cannot be handled by DB2 HPU, the data selection is done by DB2.

The setting of the DB2 parameter affects not only the data consistency but also the performance. Consider both data consistency and performance when you select the appropriate DB2 option.

The following options for the DB2 parameter are available:

FORCE

SQL access is always used.

YES Direct access is used when the SELECT statement can be directly processed by DB2 HPU. Otherwise, SQL access is used.

NO Direct access is used, or the UNLOAD statement is rejected because DB2 HPU cannot process it.

QUIESCE

When direct access is used with the unload, some data from the table might be updated, and the current data might be stored only in the DB2 buffers. To ensure that all the updates are written into the table space LDS, set the QUIESCE keyword to YES. However, new updates can occur during the unload. Specifying QUIESCE YES does not prevent new updates from happening, so this setting is not sufficient to ensure consistent data. You must specify QUIESCE YES when you use the DB2 direct access retrieval method.

LOCK To prevent concurrent access for updates, set the LOCK keyword to YES. This setting is useful for both direct and SQL access modes. When you specify LOCK YES, the table can be accessed only in read-only mode while DB2 HPU is unloading data from the table. YES is required when any access mode is selected.

You can specify the DB2, QUIESCE, and LOCK keywords in the DB2 HPU SYSIN and when you customize DB2 HPU. Use the VUU011/ULSEDB2, VUU013/ULQSCE, and VUU012/ULLOCK parameters to control the default behavior of DB2 HPU with these options.

When you specify QUIESCE, set an appropriate value for the VUU028/ULQSCEBH parameter.

Tip: To ensure data consistency, consider specifying QUIESCE YES and LOCK YES. The DB2 FORCE and LOCK YES combination also ensures data consistency, but reduces performance.

The following table shows the consistency level that is achieved when you specify different combinations of the QUIESCE, LOCK, and DB2 keywords:

Table 12. Consistency levels for QUIESCE, LOCK, and DB2 combinations

QUIESCE setting	LOCK setting	DB2 setting	Consistency level
NO	NO	NO	Not ensured (1,2)
NO	NO	YES	Not ensured (2,3)
NO	NO	FORCE	Not ensured (2)
NO	YES	NO	Not ensured (4)
NO	YES	YES	Not ensured (3)
NO	YES	FORCE	Ensured (5)
YES	NO	NO	Not ensured (2)
YES	NO	YES	Not ensured (2)
YES	NO	FORCE	Not ensured (2,6)
YES	YES	NO	Ensured
YES	YES	YES	Ensured
YES	YES	FORCE	Ensured (5,6)

Notes:

1. You can obtain data consistency by performing a QUIESCE or a STOP/START sequence against the table. However, other action might be needed to achieve consistency. Refer to notes 2 - 6 for more information.
2. Data consistency is obtained only if no DB2 update is made against the table while the unload process is in progress.
3. Data consistency depends on the SELECT statement because it can be processed in direct access mode or in SQL access mode. If direct access is selected by DB2 HPU, note 1 also applies.
4. Avoid this setting because it locks the table without ensuring that the output data is consistent.
5. Although this setting ensures consistency, it can lead to lower performance because any SELECT statement will be processed in the SQL access mode. Avoid using this setting.
6. Specifying QUIESCE YES in this case is useless because DB2 FORCE is also specified. Specify QUIESCE YES only when you specify DB2 NO or DB2 YES.

Related reference:

“DB2 HPU DB2 parameters” on page 397

The DB2 parameters section on the Product Parameters panel (CCQPPRD) in Tools Customizer contains the DB2 parameters that are used by DB2 HPU.

Chapter 4. DB2 HPU batch mode operation

You use the DB2 HPU batch utility program to unload DB2 data.

DB2 HPU uses syntax and JCL that is compatible with the standard DB2 UNLOAD utility. However, DB2 HPU does not support the entire DB2 syntax for SELECT statements. SELECT statements that are not supported are passed to DB2 for processing.

Topics:

- “Required privileges for running DB2 HPU”
- “JCL for running DB2 HPU in batch mode” on page 76
- “DB2 HPU EXEC statement” on page 76
- “Ddname allocation” on page 79
- “DB2 HPU syntax” on page 82
- “Example: The SYSPRINT data set” on page 209

Required privileges for running DB2 HPU

You must have specific user-authority privileges to run DB2 HPU.

The following privileges are required:

- When RACF[®] is used, the READ authority is not required on DB2 data sets that contain the DB2 catalog and the unloaded DB2 table spaces and indexes because DB2 HPU bypasses the RACF control. When other security products are used, such as ACF2, DB2 HPU does not bypass the control. Therefore, the READ authority is required on all the DB2 data sets that DB2 HPU accesses. The list of objects that are accessed at the VSAM level depends on the value of the VUM027/SQLACCES PARMLIB parameter for the DB2 catalog and the DB2 keyword for the unloaded table spaces and indexes.
- If an image copy is unloaded, RACF READ authority is required on the image copy data set.
- DB2 SELECT privilege is required on the tables or views to be unloaded, and DISPLAYDB privilege is required on the database.
- If QUIESCE YES is specified, you must be authorized to run the DB2 QUIESCE utility for the table space.
- If LOCK YES is specified, you must have SELECT privilege on all tables of the unloaded table space.
- If QUIESCECAT YES is specified, you must be authorized to quiesce the DB2 catalog table spaces that are shown for the VUM014/QUIESCAT parameter.
- DB2 HPU can use real-time statistics to more accurately estimate the amount of data to be processed, which improves the allocation of work data sets. To use real-time statistics, the owner of the DB2 HPU plan, which is set by the VUM012/PLANOWN parameter, must have the DB2 SELECT privilege for the SYSIBM.SYSTABLESPACESTATS and SYSIBM.SYSINDEXSPACESTATS tables. If the requested authority is not available, DB2 HPU uses a less accurate method for estimating the amount of data.

JCL for running DB2 HPU in batch mode

Sample JCL for running DB2 HPU in batch mode is provided in SINZSAMP member INZEXECU. You can copy and modify this JCL as needed for your environment.

For example, two SELECT statements have been added to the following JCL. These SELECT statements unload data from the USER01.TABLE01 table and the USER01.TABLE02 table, both of which reside in the DBNAME1.TSNAME1 table space.

```
*****
//*
//* DB2 UNLOAD JCL
//*
//* IN THIS SAMPLE :
//*
//* - THE DB2 SUBSYSTEM IS DB2P
//* - THE DB2 UNLOAD LOAD MODULES ARE
//*   IN THE LOADLIB &VIZ004
//*
//* - THE EXECUTION REPORT WILL BE
//*   WRITTEN ON THE ddname SYSPRINT
//*
*****
//STEP1 EXEC PGM=INZUTILB,REGION=0M,DYNAMNBR=99,
//      PARM='DB2P,DB2UNLOAD'
//STEPLIB DD DSN=DB2UNLOAD.LOAD,DISP=SHR
//        DD DSN=PRODDB2.DSNEXIT,DISP=SHR
//        DD DSN=PRODDB2.DSNLOAD,DISP=SHR
//*
//SYSIN DD *
        UNLOAD TABLESPACE DBNAME1.TSNAME1
        DB2 YES
        QUIESCE YES QUIESCECAT YES
        OPTIONS DATE DATE_A
        SELECT COL1,COL2 FROM USER01.TABLE01
        ORDER BY 1 , COL2 DESC
        OUTDDN (UNLDDN1)
        FORMAT VARIABLE ALL
        SELECT COL3,COL4 FROM USER01.TABLE02
        OUTDDN (UNLDDN2)
        FORMAT DSNTIAUL
        LOADDDN LOADDDN1 LOADOPT (RESUME NO REPLACE)
/*
//SYSPRINT DD SYSOUT=*
/*
//***** ddnameS USED BY THE SELECT STATEMENTS *****
/*
//UNLDDN1 DD DSN=.....,DISP=SHR
//UNLDDN2 DD DSN=.....,DISP=SHR
//LOADDDN1 DD DSN=.....,DISP=SHR
```

DB2 HPU EXEC statement

A sample EXEC statement that you use to run DB2 HPU in batch mode is provided in SINZSAMP member INZEXECU. You can copy and modify this sample as needed for your environment.

The following example shows the format of the EXEC statement:

```
EXEC PGM=module-name,PARM='subsystem-name/group-attachment-name,job-ID[,HIDDEN(YES/NO)]'
```

The EXEC statement contains the following parameters. Content in the PARM field must be enclosed in single quotation marks (').

Module-name

INZUTILB: Specify this module name to process

- a DB2 HPU native syntax
- an Unload Plus syntax
- a Fast Unload syntax.

INZUTILC: Specify this module name to process

- a DB2 UNLOAD syntax

subsystem-name/group-attachment-name

Specifies the name of the DB2 subsystem. In a non-data-sharing environment, this value is the name of a DB2 subsystem. In a SYSPLEX data-sharing environment, this value may be either a DB2 group attachment name or the name of one of the subsystems that belongs to the data sharing group. The name must be one of the SSID listed on the Tool Customizer "Associate DB2 Entry for Product".

job-ID

Specifies the unique identifier for your DB2 HPU job. Special characters are not permitted. This identifier is used as a utilid when DB2 HPU needs to invoke a DB2 utility (for example the QUIESCE utility).

If not specified, DB2 HPU launches the DB2 utilities with an utilid built with `userid.jobname` unless the Fastunload syntax is used, in which case the utilid is built with `jobname.userid`.

For concurrency purpose, make sure you do not try to run DB2 HPU jobs using the same utilid - determined as described above - against the same DB2 subsystem/data sharing simultaneously.

HIDDEN

Specifies whether hidden columns are unloaded when a SELECT * statement from a table name or from a LISTDEF statement is used. This parameter is optional. It applies only to SELECT statements that are processed natively by DB2 HPU. Use the HIDDEN keyword in the OPTIONS block to override this parameter.

NO Specifies that hidden columns are not unloaded when a SELECT * statement is used.

YES Specifies that hidden columns are unloaded when a SELECT * statement is used. For SELECT statements that are processed by DB2, the HIDDEN parameter is ignored.

If the HIDDEN parameter is not specified in the SYSIN or in the PARM field, the default value is the value that is specified for the VUU042/ULHIDDEN PARMLIB parameter.

Important: If you specify FASTUNLOAD in SYSIN to use the Fast Unload syntax, precede *subsystem_name/group_attachment_name/job_id* in the PARM field with the EP=UTLGLCTL positional parameter, as shown in the following example:

```
PARM='EP=UTLGLCTL/DSN5,,DB2UNLOAD'
```

Note: The input parm string can be specified in a file allocated to the ddname APRM or to the ddname specified in the EXEC PARM, using the syntax PARM='DDNAME=ddname' positional parameter, as shown in the following example:

```

EXEC INZUTILB,PARM='DDNAME=MYDD'
MYDD DD *
DBP, MYUID
/*

```

Processing mode considerations

Native processing versus SQL mode processing

DB2 HPU retrieves the data to be unloaded in two ways:

- either by directly accessing the data from the data sets where it is stored (either the VSAM data sets handled by DB2 or the image copy data sets),
- or by requesting DB2 to provide the data from its buffers.

The first is called the native processing (also referred to as native mode) and the latter is called the SQL processing (also referred to as SQL mode). The applicable modes depend on the kind of unload to be performed. When both modes are available for a given unload, the mode to be used can be specified via the DB2 option or its default counterpart in the DB2 HPU settings (VUU011/ULSEDB2) with the DB2 HPU syntax or with an equivalent keyword with the other syntaxes.

The table below shows which mode is applicable.

Table 13.

Unload request	Native mode	SQL mode
Physical unload (1)	YES	NO
Unload from an image copy (2)	YES	NO
Logical unload (3) with a supported SQL statement (4)	YES (5)	YES (5)
Logical unload (3) with an unsupported SQL statement (4)	NO	YES

(1) DB2 HPU UNLDDN option or equivalent feature with other syntaxes.

(2) DB2 HPU COPYDDN option or equivalent feature with other syntaxes.

(3) DB2 HPU OUTDDN option or equivalent feature with other syntaxes.

(4) DB2 HPU only supports a part of the SQL SELECT statements (see “Fast select and fast listdef select blocks syntax and description” on page 173 for more details).

(5) DB2 HPU selects the mode according to the instructions or preferences given by the DB2 option or its default counterpart in the DB2 HPU settings (VUU011/ULSEDB2).

SQL processing details

When DB2 HPU passes a SELECT statement to DB2 to retrieve the data (SQL mode), the following systematical or optional actions that might have an influence on the way DB2 processes the statement are performed.

- The FOR READ ONLY clause is appended to the SELECT statement (1).

- The USE CURRENTLY COMMITTED or WAIT FOR OUT-COME is appended to the SQL statement according to the ACCPREP specification (via the ACCPREP option in technical parameters block or the VUU061/ACCPREP parameter) (2).
- The SQL statement, with potential changes described above is run after some special registers have been set as follows:
 - The CURRENT DEGREE special register is set according to the ULDEGREE option in technical parameters block or the VUU021/ULDEGREE parameter (1)
 - The CURRENT SQLID special register is set according to the SQLID specified via the GLOBAL OPTIONS block (2).
 - The CURRENT QUERY ACCELERATION special register is set according to the QUERY_ACCELERATION option in the OPTIONS block (2).
 - The CURRENT OPTIMIZATION HINT special register is set according to the OPTIMIZATION_HINT option in the OPTIONS block (2).
- According to the UNLROWSET option or the VUU035/UNLROWSET parameter, the SQL statement is run in either single or multi-row fetch mode (1).

(1) systematical action

(2) optional action

Ddname allocation

DB2 HPU uses two types of ddnames: ddnames that DB2 HPU allocates and ddnames that you must allocate.

DB2 HPU-allocated ddnames

DB2 HPU dynamically allocates some of the ddnames that are required to run unload jobs.

The following reserved ddnames are allocated dynamically by DB2 HPU:

ICxx-ICxxnnnn

The ddname that is used to allocate the input image copy, where *xx* is a unique two-character alphanumeric value, and *nnnn* is the partition number. This ddname is used in the following conditions:

- When you specify COPYDDN LAST_IC or COPYDDN *integer*, the ddname that is used to allocate the input image copy is ICxx or ICxxnnnn, depending on whether you specified the PARTITIONED keyword or the ANYTYPE keyword.
- When you use the COPYDDN keyword with a template name, ICxx is used to allocate a global input image copy, and ICxxnnnn is used to allocate an input image copy per partition.

LBFnnnn

The ddname that is used to allocate the output file for a LOB file reference, where *nnnn* is the partition number. The same ddname is reused during the unload process to allocate all LOB file references for a specific LOB column and a specific partition.

SORT ddnames

DB2 HPU calls the sort utility when you specify an ORDER BY clause or an ORDER CLUSTER clause, which means that sort ddnames are dynamically allocated.

User-allocated ddnames

To run unload jobs, you must allocate certain ddnames in the DB2 HPU JCL.

To allocate ddnames, specify the following DD statements in your DB2 HPU JCL. Some of these DD statements are required and some are optional.

STEPLIB or JOBLIB

To use DB2 HPU, the job must have access to the DB2 HPU LOADLIB and the DB2 DSNLOAD libraries. You can give DB2 HPU access to these libraries by specifying a JOBLIB DD statement, a STEPLIB DD statement, or by including the library names in the LINKLIST.

If you do not specify the DSNEXIT library in the VZD007 PARMLIB parameter for the corresponding DB2 subsystem, you must specify the DSNEXIT library in the STEPLIB or JOBLIB libraries, and you must specify it before the DB2 DSNLOAD in the concatenation.

SYSIN

This data set contains commands for DB2 HPU.

SYSPRINT

This ddname specifies the data set that receives the report from DB2 HPU.

SYSTEMM

This ddname is optional. It receives the additional diagnostic information from running DB2 HPU.

SYSABEND

This ddname is optional. Because the INZUTILB module runs in key 7, the standard dump mechanism does not apply. If an abend occurs, a dump is produced by using the SNAP macro in the SYSABEND ddname, if this ddname is allocated in the JCL. The VZM009 parameter defines a list of system codes, such as system codes for X37 abends, for which dumps are not produced.

Other dump ddnames, such as SYSUDUMP and SYSMDUMP, are not used by DB2 HPU.

UTPRINT

This ddname is optional. It specifies the output data set for sort utility messages. If the VUX020/SORTCLAS PARMLIB parameter is set to a nonblank value, the UTPRINT ddname is not used for the sort messages, and DB2 HPU dynamically allocates one ddname for the messages of each sort that is invoked.

INFPLIB

This ddname is conditional. It connects DB2 HPU to the PARMLIB that contains the INZUTIL member.

If the INZPARM member was customized and submitted during installation, you can omit this ddname.

copydd

This DD statement is optional. Specify a DD statement with a name that matches the value of *copydd* in the COPYDDN parameter that is specified in the SYSIN DD. You can use a TEMPLATE statement instead of allocating a DD statement in the JCL.

This DD statement names the image copy data set from which the unload is to be done.

For nonpartitioned table spaces or to use a global image copy of a partitioned table space, allocate the COPYDD ddname in your JCL. You can also concatenate image copies for each partition under a single ddname to avoid using partition parallelism.

To enable DB2 HPU to process partitioned table spaces in parallel by using image copies as input, use a 1- to 7-digit sequential number to specify one *copyddnnn* statement for each partition that you want to unload. The following example shows how you might specify this DD statement:

```
copydd01  
copydd02  
copyddnnn
```

, where *nnn* is a 1- to 7-digit sequential number. DB2 HPU searches for all possible ddnames and partition numbers, with or without leading zeros. For example, if you specify COPYDDN CPY, for partition 1, DB2 HPU searches in the following order for CPY00001, CPY0001, CPY001, CPY01, and CPY1, and uses the first ddname that is found. Ddnames that are allocated for partitions can be different lengths, such as CPY1, CPY02, CPY003, and CPY00004.

outdd

This DD statement is optional. Specify a DD statement with a name that matches the value of *outdd* in the OUTDDN parameter that is specified in the SYSIN DD. You can use a TEMPLATE statement instead of a DD statement that is allocated in the JCL.

This DD statement names the data set that will contain the result of a SELECT statement (logical unload).

unldd

This DD statement is optional. Specify a DD statement with a name that matches the value of *unldd* in the UNLDDN parameter that is specified in the SYSIN DD. You can use a TEMPLATE statement instead of a DD statement that is allocated in the JCL.

This DD statement names the data set that will contain the physical unload of your table space.

loaddd

Specify a DD statement with a name that matches the value of *loaddd* in the LOADDDN parameter that is specified in the SYSIN DD. You can use a TEMPLATE statement instead of a DD statement that is allocated in the JCL.

This DD statement names the data set that will contain the SYSIN DD for a load that allows a RELOAD of a SELECT statement in DSNTIAUL or VARIABLE format into the same or a different table.

Restriction: Depending on the type of format, DB2 HPU cannot generate the RELOAD of a SELECT statement if records of variable size that do not contain headers are detected. When this situation occurs, the following conditions exist:

- The DELIMITED format is supported only in DB2 Version 8 and later releases because the DB2 LOAD utility accepts this format.
- Except for the variable columns without headers, the DSNTIAUL, VARIABLE, USER, and EXTERNAL formats are supported. DB2 HPU does not generate a LOADDDN statement when you specify LENGTHBYTE NO and LENGTH REAL with variable columns in the USER format, an INTO clause, or a REFORMAT clause.

Related reference:

“TEMPLATE block syntax and description” on page 89

Use the TEMPLATE control statement to allocate data sets without using JCL DD statements when an UNLOAD command is processed.

“DB2 HPU DB2 parameters” on page 397

The DB2 parameters section on the Product Parameters panel (CCQPPRD) in Tools Customizer contains the DB2 parameters that are used by DB2 HPU.

“SELECT block syntax and description” on page 168

The SELECT statement specifies that a logical unload is to be done and indicates the parameters that are associated with the unload job. The SELECT block is a part of the UNLOAD block.

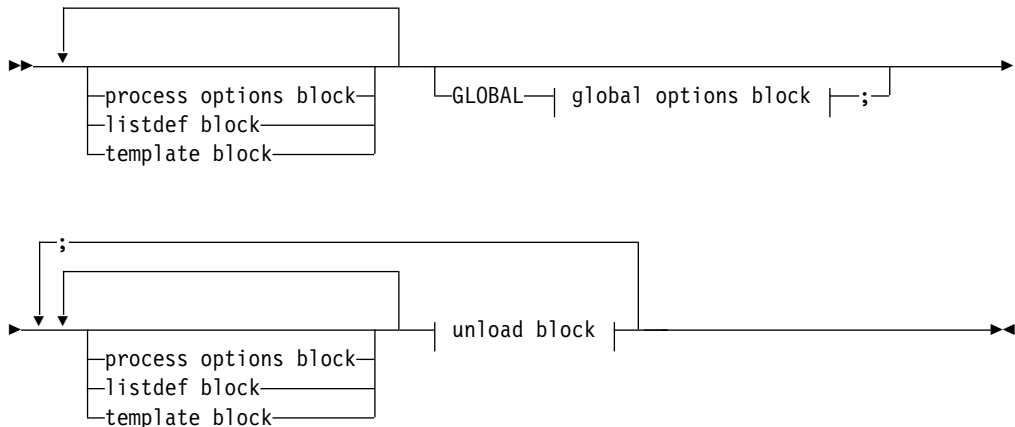
“UNLOAD block syntax and description” on page 110

Use the UNLOAD statement to specify the data to be unloaded from a table space or an image copy. The UNLOAD block contains the OPTIONS and SELECT blocks.

DB2 HPU syntax

The DB2 HPU syntax is divided into five major blocks: PROCESS OPTIONS, LISTDEF, TEMPLATE, GLOBAL OPTIONS, and UNLOAD. Only the UNLOAD block is required. All other major blocks are optional.

The following syntax diagram describes the DB2 HPU syntax and how to code the different major syntax blocks.



You can specify the PROCESS OPTIONS block, LISTDEF block, and TEMPLATE block either before the GLOBAL OPTIONS block or before each UNLOAD command. LISTDEF and TEMPLATE definitions are processed sequentially and can be used in all UNLOAD commands that follow their definition.

You cannot redefine a TEMPLATE with a name that is already defined in the same SYSIN. However, you can redefine a TEMPLATE in the SYSIN that is defined in a TEMPLATE library.

The DB2 HPU parsers accepts comment lines that begin with two dashes (##) in column one.

Comment lines that begin with an asterisk (*) in column one are also accepted when they are coded before the first keyword of the SYSIN.

Topics:

- “PROCESS OPTIONS block syntax and description”
- “LISTDEF block syntax and description” on page 84
- “TEMPLATE block syntax and description” on page 89
- “GLOBAL OPTIONS block syntax and description” on page 105
- “UNLOAD block syntax and description” on page 110

PROCESS OPTIONS block syntax and description

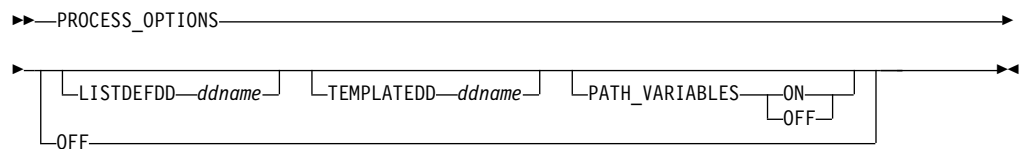
Use the PROCESS OPTIONS control statement to change and restore the default ddnames that DB2 HPU uses for LISTDEF definition libraries and TEMPLATE definition libraries.

The PROCESS OPTIONS control statement corresponds to a subset of functions in the OPTIONS control statement in the DB2 utilities. To avoid confusing the OPTIONS keyword that exists in the DB2 HPU syntax with the OPTIONS control statement in the DB2 utilities, the equivalent keyword in DB2 HPU is PROCESS_OPTIONS.

The LISTDEFDD and TEMPLATEDD parameters can point to an existing LISTDEF definition library or to a TEMPLATE definition library that is used by the standard DB2 utilities. Any unsupported keywords in the LISTDEF definitions or TEMPLATE definitions are ignored.

You must specify at least one keyword after you specify PROCESS_OPTIONS. The following diagram shows the syntax of the PROCESS_OPTIONS block:

PROCESS OPTIONS block



LISTDEFDD *ddname*

Specifies the ddname of the LISTDEF definition library. A LISTDEF definition library is a data set that contains only LISTDEF control statements. This data set is processed only when a referenced LIST is not found in the SYSIN.

The default value is SYSLISTD.

TEMPLATEDD *ddname*

Specifies the ddname of the TEMPLATE definition library. A TEMPLATE library is a data set that contains only TEMPLATE control statements. This data set is processed only when a referenced name does not exist in the job step as a ddname and is not found in the SYSIN as a TEMPLATE name.

The default value is SYSTEMPL.

PATH_VARIABLES

Specifies whether template variables are substituted in path names.

ON Specifies that template variables are substituted.

OFF Specifies that template variables are not substituted. Specified path names are used as they are coded.

The default value is ON.

OFF Specifies that all default options are to be restored. You cannot specify any other PROCESS_OPTIONS keywords if you specify PROCESS_OPTIONS OFF. Specifying PROCESS_OPTIONS OFF is equivalent to specifying OPTIONS LISTDEFDD SYSLISTD TEMPLATEDD SYSTEMPL PATH_VARIABLES ON.

Related reference:

“LISTDEF block syntax and description”

Use the LISTDEF control statement to group table spaces into reusable lists and to specify these lists in UNLOAD commands or in SELECT statements to indicate that all the items in the lists are to be processed.

“TEMPLATE block syntax and description” on page 89

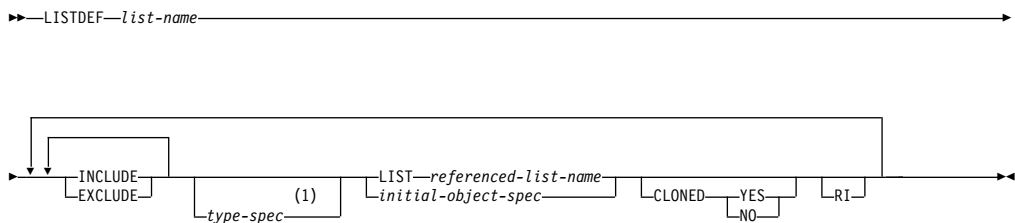
Use the TEMPLATE control statement to allocate data sets without using JCL DD statements when an UNLOAD command is processed.

LISTDEF block syntax and description

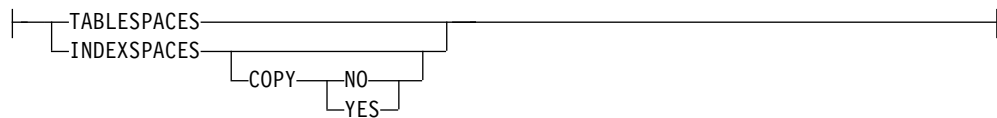
Use the LISTDEF control statement to group table spaces into reusable lists and to specify these lists in UNLOAD commands or in SELECT statements to indicate that all the items in the lists are to be processed.

The following diagram shows the syntax of the LISTDEF block.

LISTDEF block



type spec:



initial object spec:

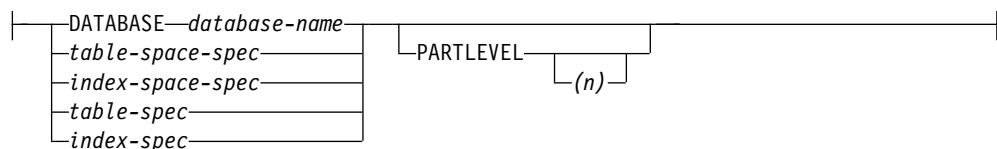
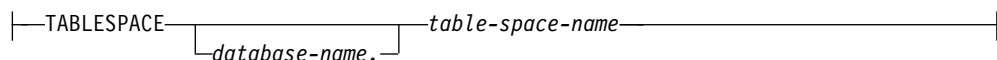


table space spec:



INDEXSPACES COPY YES|NO

Specifies that the INCLUDE or EXCLUDE clause will create a list of related index spaces. INDEXSPACES is accepted only for compatibility with the syntax of the LISTDEF statement in the DB2 utilities. DB2 HPU cannot process lists of index spaces.

LIST *referenced-list-name*

Specifies the name of a previously defined object list that will be expanded and used for the initial search for the object. You must explicitly specify the *referenced-list-name* name. You cannot specify pattern-matching characters, such as %, *, ?, and _ , for lists. You can use the LIST keyword to make aggregate lists of lists, to exclude entire lists from other lists, and to develop lists of objects that are related to other lists.

DATABASE *database-name*

Specifies the database that will be used for the initial search for the object. You can specify the database name explicitly or as a pattern-matched name. DATABASE * and DATABASE % are not supported. If you specify DATABASE, you must also specify either TABLESPACES or INDEXSPACES as the list type. Depending on the list type that you specify, DB2 HPU includes all table spaces or index spaces in the database name that satisfy the pattern-matching expression in the list. You cannot specify DSNDB01, DSNDB06, DSNDB07, or user-defined work file databases in a LISTDEF definition.

TABLESPACE *database-name.table-space-name*

Specifies the table space that will be used for the initial search for the object. If you specify TABLESPACE, the default list type is TABLESPACES. All table spaces that satisfy the pattern-matching expression are included in the list unless the list is modified by other keywords. TABLESPACE *.* and TABLESPACE %.% are not supported.

Use *database-name* to specify the name of the database to which the table space belongs. The default value is DSNDB04. Use *table-space-name* to specify the name of the table space. You can explicitly specify *database-name*, *table-space-name* or both, or you can use pattern-matching characters to specify these values. You cannot include any objects in DSNDB07 or any user-defined work file databases in a LISTDEF definition. Pattern matching is not supported for DSNDB01 and DSNDB06 objects.

LOB and XML table spaces are not selected when they match a LISTDEF definition that was processed by DB2 HPU.

INDEXSPACE *database-name.index-space-name*

Specifies the index space that will be used for the initial object search. If you specify INDEXSPACE, the default list type is INDEXSPACES. All index spaces that satisfy the pattern-matching expression are included in the list unless the index spaces are excluded by other LISTDEF options. INDEXSPACE *.* and INDEXSPACE %.% are not supported.

Use *database-name* to specify the name of the database to which the index space belongs. The default value is DSNDB04. Use *index-space-name* to specify the name of the index space. You can explicitly specify *database-name*, *index-space-name* or both, or you can use pattern-matching characters to specify these values. You cannot include any objects in DSNDB07 or any user-defined work file databases in a LISTDEF definition. Pattern matching is not supported for DSNDB01 and DSNDB06 objects.

TABLE *creator-ID.table-name*

Specifies the table that will be used for the initial search for the object. If

you specify TABLE, the default list type is TABLESPACES. All table spaces that contain tables that satisfy the pattern-matching expression are included in the list unless the list is modified by other keywords. TABLE *.* and TABLE %.% are not supported.

Use *creator-ID* to specify the qualifier creator ID for the table. The default value is the user identifier for the utility. Use *table-name* to specify the name of the table. If you specify a table name and the CLONED keyword, the CLONED keyword is ignored. You can explicitly specify or use pattern-matching characters to specify *creator-ID*, *table-name*, or both. However, the underscore (_) pattern-matching character is ignored in a table name. Pattern matching is not supported for catalog and directory objects. When you include a catalog and directory objects in a LISTDEF statement, you must use their fully qualified names. If the name contains a blank, enclose the table name in quotation marks.

INDEX *creator-ID.index name*

Specifies the index that will be used for the initial search for the object. If you specify INDEX, the default list type is INDEXSPACES. All index spaces that contain indexes that satisfy the pattern-matching expression are included in the list unless the list is modified by other keywords. INDEX *.* and INDEX %.% are not supported.

Use *creator-ID* to specify the qualifier creator ID for the index. The default is the user identifier for the utility. Use *index-name* to specify the name of the index. You can explicitly specify *creator-ID*, *index-name*, or both, or you can use pattern-matching characters to specify these values. However, the underscore pattern-matching character is ignored in an index name. Pattern matching is not supported for catalog and directory objects. When you include a catalog and directory objects in a LISTDEF statement, you must use their fully qualified names. If the name contains a blank, enclose the index name in quotation marks.

PARTLEVEL

Specifies the partition granularity for partitioned table spaces. You cannot specify the PARTLEVEL keyword with the RI keyword. The integer partition number (*n*) is greater than or equal to zero ($n \geq 0$). If you specify PARTLEVEL 0, the resulting list contains one entry for each nonpartitioned object. If you specify PARTLEVEL with a nonzero operand, the resulting list contains one entry for the specified partition for partitioned objects and one entry for each nonpartitioned object. If you specify PARTLEVEL without specifying the integer part number, the resulting list contains one entry for each partitioned object and one entry for each nonpartitioned object. DB2 HPU supports this option only for syntax compatibility with the DB2 utilities. Using PARTLEVEL without specifying the integer part number *n* does not mean that data is unloaded per partition. You can remove an INCLUDE clause with the PARTLEVEL keyword from the list only by using an EXCLUDE clause with the PARTLEVEL keyword.

Parentheses are not required around the partition number when you specify the PARTLEVEL keyword.

CLONED

Specifies that the INCLUDE or EXCLUDE clauses will return only the names of cloned tables, table spaces that contain cloned tables, indexes on cloned tables, or index spaces that contain indexes on cloned tables. If you specify CLONED, other keywords in the LISTDEF statement that refer to related objects refer to the clones of those objects. If you also specify a table name, the CLONED keyword is ignored.

RI Specifies that all objects that are referentially related to the object expression (PRIMARY KEY <--> FOREIGN KEY) will be included in the list. DB2 HPU processes all referential relationships repeatedly until the entire referential set is developed. You cannot specify RI with PARTLEVEL(*n*).

Related reference:

“PROCESS OPTIONS block syntax and description” on page 83

Use the PROCESS OPTIONS control statement to change and restore the default ddnames that DB2 HPU uses for LISTDEF definition libraries and TEMPLATE definition libraries.

“Partition processing with LISTDEF and PARTLEVEL”

In DB2 HPU, a LISTDEF definition that includes the PARTLEVEL keyword without a partition number is supported only for compatibility purposes with the IBM LISTDEF utility and has no impact on how DB2 HPU unloads the data. LISTDEF definitions such as these do not activate partition parallelism.

LISTDEF specification

When you specify LISTDEF, the LISTDEF that you specify depends on whether the unload is a physical unload or a logical unload.

For physical unloads (UNLDDN), specify LIST(*list-name*) in the UNLOAD TABLESPACE command instead of specifying the table space name. A physical unload is generated for each table space that is contained in the generated list of table spaces. If the UNLDDN uses a template, a separate data set is allocated for each table space that was unloaded.

For logical unloads (SELECT + OUTDDN), specify LIST(*list-name*) in the FROM clause of the SELECT statement, and do not specify a table space name in the UNLOAD TABLESPACE command. A SELECT statement is generated for each table of each table space in the list. If a template is used for the OUTDDN and for the LOADDDN, a separate file is dynamically allocated for each table of each table space in the generated list.

Partition processing with LISTDEF and PARTLEVEL

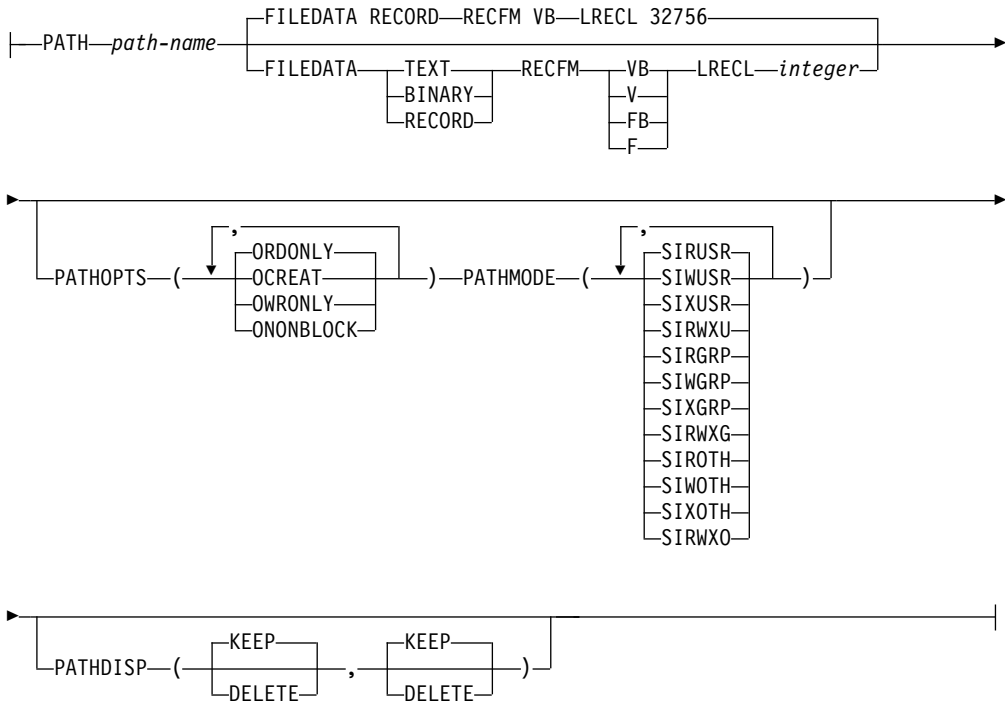
In DB2 HPU, a LISTDEF definition that includes the PARTLEVEL keyword without a partition number is supported only for compatibility purposes with the IBM LISTDEF utility and has no impact on how DB2 HPU unloads the data. LISTDEF definitions such as these do not activate partition parallelism.

When DB2 HPU writes in a separate file per partition, partition parallelism is activated automatically. When DB2 HPU writes to a single output file, partition parallelism can be forced by using the PARALLELISM keyword of the UNLOAD command or by setting the VUU036/GBLPARAL PARMLIB parameter to YES.

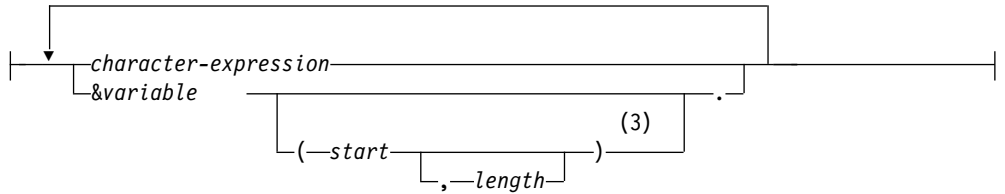
In DB2 HPU, when the list that is generated by a LISTDEF control statement contains a partitioned table space, and if the DSNAME that is defined in the TEMPLATE statement contains the &PART variable, the unloaded rows are written in a separate file for each partition. Otherwise, all rows that are unloaded from the partitioned table space are written in a single file. This condition applies to output files that are generated by the UNLDDN or OUTDDN keywords.

When the list that was generated by a LISTDEF control statement is used, the PART keyword in the UNLOAD command is ignored, and the PART keyword in the SELECT statement is not valid. You can select partitions by using the PARTLEVEL keyword in the LISTDEF statement. All partitions of the same table

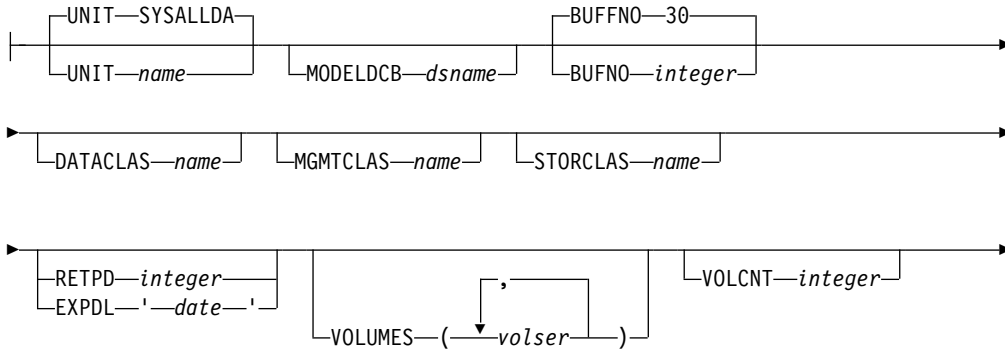
path expression:

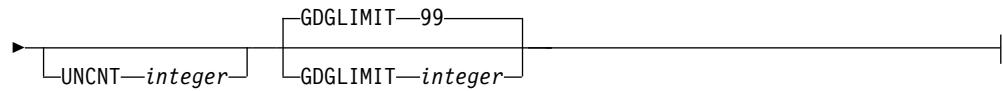


qualifier expression:



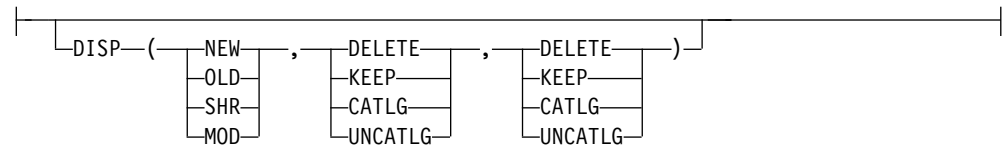
common options:



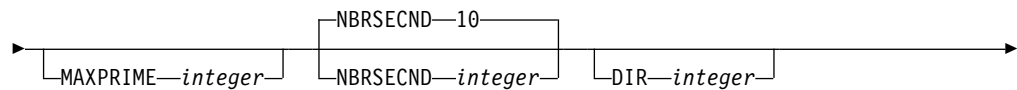
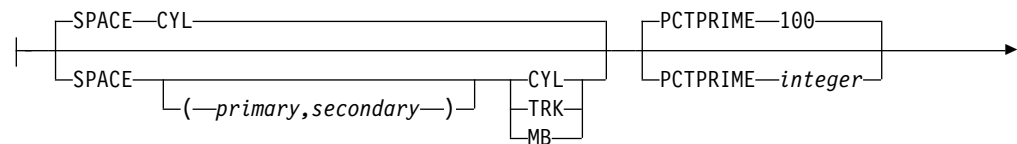


Notes:

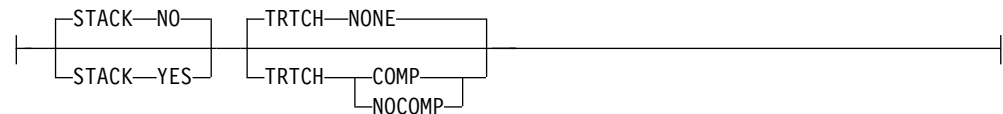
- 1 SUBSYS and *path-expression* are mutually exclusive.
- 2 The entire name expression represents one character string and cannot contain any blanks.
- 3 If you use substringing notation, the entire DSN operand must be enclosed in single quotation marks, such as 'P&PA(4,2).'



disk options:



tape options:



Important: To share TEMPLATE libraries with other DB2 utilities, DB2 HPU accepts, but ignores, all keywords that are not specified in the syntax diagram.

TEMPLATE *template-name*

Defines a data set allocation template and assigns a name to the template for subsequent reference in an UNLOAD command. The name is followed

by keywords that control the allocation of tape and disk data sets. You cannot specify both disk options and tape options in the same TEMPLATE statement. The UNIT keyword specifies a generic unit name that is already defined on your system.

Valid values are 1 - 8 alphanumeric characters and must begin with an alphabetic character.

DSN *name-expression*

Specifies the TEMPLATE for the z/OS data set name. You can specify the data set name by using symbolic variables, non-variable alphanumeric or national characters, or any combination of these characters. The name must adhere to the z/OS data set naming rules, including those rules about name length, valid characters, name structure, and qualifier length.

Data set names consist of a series of qualifiers and qualifier expressions that are separated by a period and an optional expression in parentheses. Embedded blanks are restricted. If the DSN name operand contains any special characters, it must be enclosed in single quotation marks. For example, in the following TEMPLATE statement, the DSN operand contains parentheses as special characters, so the entire operand is enclosed in single quotation marks:

```
TEMPLATE X DSN 'A.GDG.VERSION(+1)'
```

Parentheses around the DSN name operand are optional. The following DSN specification shows a DSN name operand in parentheses:

```
DSN(&DB..&TS..D&DATE.)
```

PATH *path-name*

Specifies a z/OS UNIX System Services (USS) file path name. This path name can be a USS pipe path name. The path name must be in single-byte EBCDIC format and must not exceed 255 bytes. If the path name contains blanks, you must enclose it in single quotation marks.

DB2 HPU can use a template with the PATH keyword only for the following types of files:

- UNLDDN for output of physical unloads
- OUTDDN for output of logical unloads
- LOADDDN for the LOAD SYSIN for logical unloads
- DDLDDN to specify the DDL for unloading from an image copy
- A template that is used to define a LOB file reference when LOB or XML data is unloaded

You can use a variable name for a complete directory name or part of a directory name. Before the files are generated, DB2 HPU ensures that the directory exists. DB2 HPU does not create the directory. If the directory does not exist, DB2 HPU issues an error.

Syntax rules regulate how DB2 HPU interprets variable names. When you use an ampersand (&), DB2 HPU interprets the ampersand as the introduction to a variable name. To generate an ampersand in the path name, you must use a backslash (\) as an escape character before the ampersand. In the following example, &TS. is coded as a variable name, and &SPECIAL is coded as regular text.

```
PATH '/u/&TS.\&SPECIAL'
```

After variables have been replaced, the following path name is generated:

/u/TS01&SPECIAL

To create a USS pipe file, you must specify DSNTYPE PIPE in the TEMPLATE statement.

FILEDATA

Specifies the content type of a z/OS USS file. Valid values are TEXT, BINARY, and RECORD.

The default value is RECORD, and RECFM VB and LRECL 32756 are assumed.

RECFM

Specifies the record format of the z/OS USS file. The RECFM option is required when FILEDATA is specified. The following values are valid:

- F (Fixed)
- FB (Fixed block)
- V (Variable)
- VB (Variable block)

If FILEDATA is not specified, RECFM VB is assumed with FILEDATA RECORD LRECL 32756.

LRECL

Specifies the record length of the z/OS USS file. The LRECL option is required when FILEDATA is specified.

If FILEDATA is not specified, LRECL 32756 is assumed with FILEDATA RECORD RECFM VB.

PATHOPTS

Specifies the access and status for the z/OS USS file that is named in the PATH parameter. The following values are valid:

ORDONLY

Specifies that the utility will open the file only for read access.

OCREAT

Specifies that files are created based on the following conditions:

- If the file does not exist, the system creates it.
- If a directory that is specified in the path name does not exist, the directory and the new file are not created.
- If the file exists, the existing file is used by the utility.

OWRONLY

Specifies that the utility will open the file only for write access.

ONONBLOCK

Specifies that an open() function will return or block (wait) based on conditions for the following file types:

For a first-in first-out (FIFO) special file, such as a USS pipe file:

- If ONONBLOCK is specified with ORDONLY access, an open() function for only reading returns a file descriptor (handler) without delay.
- If ONONBLOCK is not specified with ORDONLY access, an open() function for only reading waits until a process opens the file for writing.

- If ONONBLOCK is specified with OWRONLY access, an open() function for only writing returns an error if a process does not currently have the file open for reading.
- If ONONBLOCK is not specified with OWRONLY access, an open() function for only writing waits until a process opens the file for reading.

For a character special file that supports nonblocking open:

- If ONONBLOCK is specified, an open() function returns a file descriptor without waiting until the device is ready or available. The device response depends on the type of device.
- If ONONBLOCK is not specified, an open() function waits until the device is ready or available.

Specifying ONONBLOCK does not affect other file types.

The default values are OCREAT and OWRONLY.

PATHMODE

Specifies the file mode of the HFS file. The following values are valid:

SIRUSR

Allows the file owner to read the file.

SIWUSR

Allows the file owner to write the file.

SIXUSR

If the file is a directory, allows the file owner to search; if the file is not a directory, allows the file owner to run the program in the file.

SIRWXU

If the file is a directory, allows the file owner to read, write, and search; if the file is not a directory, allows the file owner to read, write, and run the file. Specifying SIRWXU is equivalent to specifying SIRUSR, SIWUSR, and SIXUSR together.

SIRGRP

Allows users in the file group class to read the file.

SIWGRP

Allows users in the file group class to write the file.

SIXGRP

If the file is a directory, allows users in the file group class to search; if the file is not a directory, allows users in the file group class to run the program in the file.

SIRWXG

If the file is a directory, allows users in the file group class to read, write, and search; if the file is not a directory, allows users in the file group class to read, write, and run the file. Specifying SIRWXG is equivalent to specifying SIRGRP, SIWGRP, and SIXGRP together.

SIROTH

Allows users in the file other class to read the file.

SIWOTH

Allows users in the file other class to write the file.

SIXOTH

If the file is a directory, allows users in the file other class to search; if the file is not a directory, allows users in the file other class to run the program in the file.

SIRWXO

If the file is a directory, allows users in the file other class to read, write, and search; if the file is not a directory, allows users to read, write, and run the file. Specifying SIRWXO is equivalent to specifying SIROTH, SIWOTH, and SIXOTH together.

The default value is SIRUSR.

PATHDISP

Specifies the disposition of a z/OS USS file when the job step ends. Valid values are KEEP and DELETE.

The MVS system default is KEEP,KEEP.

SUBSYS *name*

Specifies the MVS BATCHPIPES SUBSYSTEM name. The specified name must be a valid BATCHPIPES SUBSYSTEM name and must be 1 - 8 characters. When SUBSYS is specified, LRECL and RECFM are required, and TEMPLATE keywords that are not compatible with SUBSYS, such as UNIT, are ignored.

Restriction: When you use BATCHPIPES and TEMPLATE with the SUBSYS keyword, the utility cannot be restarted, and the LOAD DISCARDN keyword is not supported.

LRECL *integer*

Specifies the record length of the MVS BATCHPIPES SUBSYSTEM file. LRECL is required when SUBSYS is specified.

RECFM

Specifies the record format of the MVS BATCHPIPES SUBSYSTEM file. RECFM is required when SUBSYS is specified. Valid values are F, FB, V, or VB.

character-expression

Specifies the data set name or part of the data set name by using nonvariable alphanumeric or national characters.

&variable

Specifies the data set name or part of the data set name by using nonvariable alphanumeric or national characters.

In addition to the list of variables, you can use user-defined variables in the DSN expression. You set the values for these user-defined variables by using the TEMPLATESET keyword in the OPTIONS block. Syntax rules for such user-defined variables are identical to the rules for standard variables. To avoid duplicate data set names when you specify a TEMPLATE statement with a LISTDEF list that contains a multi-table table space, specify TEMPLATESET and use a user-defined variable that corresponds with the :SELNUM predefined variable in the DSN expression.

Each symbolic variable is substituted with its related value at run time to form a specific data set name. When you use substitution variables in a DSN expression, they begin with an ampersand sign (&) and end with a period (.), as shown in the following example:

```
DSN &DB..&TS..D&JDATE..P&PART.
```

If you use only numeric variables, an invalid data set qualifier for all numeric-type variables, such as all date or time-type variables, &SEQ, and &PART, is generated. These variables must be preceded by character constants to form valid DSN qualifiers. The following specifications are valid:

P&PART.
D&DATE.

Some variables inherit default values when their values are not known. For example, the default value of &PART becomes 00000 for nonpartitioned objects.

You can also use substring notations for the data set name. This notation can prevent the data set name from exceeding the 44-character maximum. If you use substring notation, you must enclose the entire DSN operand in single quotation marks. To specify a substring, use the form &variable(start), or &variable(start,length).

The following table contains a list of JOB variables and their descriptions:

Table 14. JOB variables and descriptions

Variable	Description
&JOBNAME. or &JO.	The z/OS job name.
&STEPNAME. or &ST.	The z/OS step name. This variable might be needed if data set names from two different job steps conflict.
&USERID. or &US.	The user ID of the person who is running DB2 HPU. Valid values are 1 - 8 characters.
&UTILID. or &UT.	The utility ID truncated to eight characters and checked for invalid DSN characters.
&SSID. or &SS.	Subsystem ID for non-data-sharing environments or group attach name for data sharing environments.

The following table contains a list of UTILITY variables and their descriptions:

Table 15. UTILITY variables and descriptions

Variable	Description
&UTILNAME. or &UN.	The utility name. The value is set to HPU.
&SEQ. or &SQ.	The sequence number of the list item.

The following table contains a list of OBJECT variables and their descriptions:

Table 16. OBJECT variables and descriptions

Variable	Description
&LIST. or &LI.	The name of the list that is defined by using the LISTDEF control statement and that is referenced on the same control statement as this TEMPLATE.
&DB.	The database name.
&TS.	The table space name.
&SN.	The table space name.
&PART. or &PA.	The five-digit partition number, padded with leading zeros.

Notes:

- To generate one output file per partition, use the &PA. variable when you process LISTDEF lists that include partitioned table spaces.
- When you specify the &TS. or &SN. variables in a TEMPLATE statement that is used by an UNLOAD statement with BLOBF, CLOBF, or DBCLOBF columns, DB2 HPU substitutes the name of the table space that stores the LOB column value, not the base table space name. This substitution enables DB2 HPU to generate unique data set names for each LOB column with partitioned table spaces.
- When the SELECT statement in which the TEMPLATE is used is processed by using SQL (DB2 FORCE or DB2 YES with an unsupported SELECT statement), the &PART variable is set to one of the following values based on whether SQLPART is used:
 - The partition number when the SQLPART keyword is used
 - 00000 when the SQLPART keyword is not used

The &DB, &TS, and &SN variables are set to DB*n*, TS*n*, LOB*n* or XML*n*, depending on the table space type, where *n* is a sequential number. This sequential number ensures unity, but it does not have a specific meaning regarding the actual database or the table space that it represents.

- In DB2 HPU V3.1, when a partitioned table space was unloaded from a global full image copy (FIC) that used a template with a DSNNAME that contains the &PART variable, a single output file with partition number set to 00000 was generated. In DB2 HPU V3.2, an output file per partition with the &PART variable set to the partition number is generated.

The following table contains a list of DATE and TIME variables and their descriptions:

Table 17. DATE and TIME variables and descriptions

Variable	Description
&DATE. or &DT.	YYYYDDD
&TIME. or &TL.	HHMMSS
&JDATE. or &JU.	YYYYDDD
&YEAR. or &YE.	YYYY portion of &DATE.
&MONTH. or &MO.	MM
&DAY. or &DA.	DD
&JDAY. or &JD	DDD portion of &DATE.
&HOUR. or &HO.	HH portion of &TIME.
&MINUTE. or &MI.	MM portion of &TIME.
&SECOND. or &SC.	SS portion of &TIME.
&UNIQ. or &UQ.	Unique eight characters that DB2 HPU derives from the system clock. This set of characters begins with an alphabetic character and is followed by seven alphabetic or numeric characters.

Attention: All date and time values are set by using the STCK instruction and reflect the date and time value in Greenwich Mean Time (GMT). DATE and TIME values are captured during the initialization phase of DB2 HPU and remain constant during the STEP.

start Specifies the substring starting byte location within the current variable base value at run time.

length Specifies the length of the substring. If you specify *start* but do not specify *length*, the default value for *length* is the number of characters from the start character to the last character of the variable value at run time. For example, given a five-digit base value, &PART(4). specifies the fourth and fifth digits of the value. The *length* value must be an integer that does not cause the substring to extend beyond the end of the base value.

parenthetical-expression

Specifies part of the data set name by using nonvariable alphanumeric or national characters that are enclosed in parentheses. For example, the following expressions are valid: Q1.Q2.Q3(*member*) and Q1.Q2.Q3(+1).

UNIT Specifies the device number, device type (generic), or group name for the data set. The default value is SYSALLDA.

MODELDCB *dsname*

Specifies the name of the data set on which the TEMPLATE statement is based. DCB information is read from this model data set.

When the template corresponds to a GDG, MODELDCB uses the VUX033/TMPLDSCB PARMLIB parameter as the default value.

BUFNO *integer*

Specifies the number of BSAM buffers. Valid values are 0 - 99. The default value is 30.

DATACLAS *name*

Specifies the SMS data class. The *name* value must be a valid SMS data class and must be 1 - 8 characters. If you specify DATACLAS, the data set is cataloged. If you do not specify DATACLAS, no SMS DATACLAS is specified.

MGMTCLAS *name*

Specifies the SMS management class. The *name* value must be a valid SMS management class and must be 1 - 8 characters. If you specify MGMTCLAS, the data set is cataloged. If you do not specify MGMTCLAS, no MGMTCLAS is specified to SMS.

STORCLAS *name*

Specifies the SMS storage class. The *name* value must be a valid SMS storage class and must be 1 - 8 characters. If you specify STORCLAS, the data set is cataloged. If you do not specify STORCLAS, no STORCLAS is specified to SMS.

RETPD *integer*

Specifies the retention period in days for the data set. Valid values are 0 - 9999. If you specify DATACLAS, MGMTCLAS, or STORCLAS, the class definition might control the retention. RETPD cannot be specified with EXPDL.

EXPDL '*date*'

Specifies the expiration date for the data set, in the form YYYYDDD, where YYYY is the 4-digit year, and DDD is the 3-digit Julian day. The '*date*' value must be enclosed in single quotation marks. If you specify DATACLAS, MGMTCLAS, or STORCLAS, the class definition might control the retention. EXPDL cannot be specified with RETPD.

VOLUMES (*volume-1,volume-2,...*)

Specifies a list of volume serial numbers for this allocation. The specified

number of volumes cannot exceed the specified value or default value of the VOLCNT keyword. The first volume must contain enough space for the primary space allocation. If an individual volume serial number contains leading zeros, it must be enclosed in single quotation marks.

VOLCNT (*integer*)

Specifies the maximum number of volumes that an output data set might require. Valid values are 0 - 255.

Unless a tape storage class is used, the default value for tape templates is 95. If a tape storage class is used, the default value is the value that is set by the tape storage class.

DB2 HPU does not set a default value for disk templates. Operating system defaults apply.

UNCNT (*integer*)

Specifies the number of devices that are to be allocated. Valid values are 0 - 59. If UNIT specifies a specific device number, the value of UNCNT must either be 1, or it must be omitted.

GDGLIMIT (*integer*)

Specifies the number of entries that are to be created in a GDG base if a GDG DSN is specified and the base does not exist. If a GDG base does not exist and you do not want to define one, specify GDGLIMIT (0). Valid values are 0 - 255.

The default value is 99.

DISP (*status, normal-termination, abnormal-termination*)

Specifies the data set disposition. You must specify values for all of the following variables:

status Specify one of the following standard z/OS values: NEW, OLD, SHR, and MOD.

normal-termination

Specify one of the following standard z/OS values: DELETE, KEEP, CATLG, and UNCATLG.

abnormal-termination

Specify one of the following standard z/OS values: DELETE, KEEP, CATLG, and UNCATLG.

Default values for the disposition are NEW,CATLG,CATLG for output files for TEMPLATE statements that UNLDDN, OUTDDN, or LOADDN statements, and SHR,KEEP,KEEP for input files for TEMPLATE statements that use COPYDDN or DDLDDN statements. When a PDS or PDSE is created by a TEMPLATE statement, its default disposition is NEW,CATLG,CATLG. When new members are created in the PDS or PDSE, the disposition is SHR,KEEP,KEEP.

SPACE (*primary,secondary*)

Specifies the z/OS disk space allocation parameters. If you specify (*primary,secondary*) values, these values are used instead of the values that are calculated DB2 HPU. When specifying primary and secondary quantities, you must either specify both values or omit both values. Use the MAXPRIME option to set an upper limit on the primary quantity.

Valid values are 1 - 1677215

CYL Specifies that allocation quantities, if present, are to be expressed in cylinders and that allocation is to occur in cylinders. If SPACE CYL

is specified, without (*primary,secondary*), the DB2 HPU-calculated quantities are allocated in cylinders by using 3390 quantities for byte conversion.

- TRK** Specifies that, in the absence of values for (*primary,secondary*), the DB2 HPU-calculated quantities are to be allocated in tracks by using 3390 disk drive quantities for byte conversion. If the amount calculated is greater than one cylinder, the TRK keyword is ignored, and the data set is allocated in cylinders (CYL).
- MB** Specifies that if allocation quantities are present, they are to be expressed in megabytes, and that allocation is to occur in records. One MB is 1048576 bytes. If SPACE MB is specified without (*primary,secondary*), the DB2 HPU-calculated quantities are allocated in records by using the average record length for the data set.

The default value is CYL.

Consider the following settings and keywords for default SPACE values for allocating NEW data sets:

Default SPACE values for allocating NEW data sets

DB2 NO or DB2 YES with supported SELECT statements

When DB2 HPU uses a TEMPLATE statement to allocate an output file (using UNLDDN or OUTDDN), the space that is allocated is calculated by using the catalog statistics for the unloaded object. If there are no statistics, DB2 HPU uses the physical characteristics of the underlying data set to compute the space for the output files. If necessary, you can override the calculated space allocation by using the SPACE parameter of the TEMPLATE definition.

DB2 FORCE or DB2 YES with unsupported SELECT statements

Use the SPACE keyword with specification of primary and secondary allocation in the TEMPLATE statement, or an error message is issued.

RLSE keyword

The RLSE (release) keyword is always used to allocate new data sets so that unused space is released on deallocation except when the allocated data set is a PDS or a PDSE (using DSNTYPE PDS or LIBRARY parameters).

PCTPRIME integer

Specifies the percentage of the estimated required space that will be obtained as the primary quantity. Use the MAXPRIME keyword to set the upper limit of this value for large objects.

The default value is 100.

MAXPRIME integer

Specifies the maximum allowable primary space allocation. The value is expressed in cylinders (CYL). This value constrains the primary space value, the PCTPRIME calculation, and the size of each secondary allocation.

NBRSECND integer

Specifies the division of secondary space allocations. After the primary space is allocated, an amount of space equal to the estimated required space is divided into the specified number of secondary allocations. Valid values are 1 - 10.

The default value is 10.

DIR *integer*

Specifies the number of 256-byte records that are to be allocated for the directory of a new partitioned data set. If you are allocating a new partitioned data set, you must specify this keyword. If the TEMPLATE is being used in an UNLOAD statement with BLOBF, CLOBF, or DBCLOBF, and you specify a DSNTYPE of LIBRARY or PDS but you do not specify DIR, DB2 HPU calculates the number of 256-byte records to allocate by dividing the estimated number of records by 20.

DSNTYPE

Specifies the type of data set to be allocated.

LIBRARY

Specifies that a partitioned data set extended (PDSE) will be allocated.

PDS Specifies that a partitioned data set (PDS) will be allocated.

HFS Specifies that a hierarchical file system (HFS) file will be allocated. An HFS file is allowed only when a TEMPLATE is used to allocate a LOB file reference (CLOBF, BLOBF, or DBCLOBF columns).

NULL Specifies a null file. Use this value for a TEMPLATE statement with UNLOAD CLOBF, BLOBF, or DBCLOBF columns to unload a null LOB value. In this case, the unload data set contains a null file name.

PIPE Specifies a USS pipe file. Specify PIPE only when you allocate a new pipe output file by using the PATH option in the TEMPLATE statement. You cannot use DSNTYPE PIPE when you allocate input files, existing pipe files, standard z/OS files with the DSN option, or a LOB file reference.

BASIC

Specifies that a basic data set will be allocated. Basic format data sets are sequential data set that are specified as neither extended-format nor large-format. Basic format data sets have a size limit of 65 535 tracks (4369 cylinders) per volume.

LARGE

Specifies that a large data set will be allocated. Large format data sets are sequential data sets that can grow beyond 65 535 tracks (4369 cylinders) up to 16.777.215 tracks per volume.

EXTREQ

Specifies that an extended format data set will be allocated. The allocation should fail if that is not possible.

EXTPREF

Specifies that allocation is preferred to be in extended format, but if the necessary system resources for extended are not available, then the data set should be allocated as BASIC.

If you omit DSNTYPE, the type of data set is determined by other data set attributes, the data class for the data set, or an installation default.

STACK

Specifies whether output data sets are to be stacked contiguously on the same tape volumes.

- NO** Specifies that output data sets are not stacked contiguously on tape.
- YES** Specifies that output data sets are stacked as successive files on one tape volume. A logical tape volume can consist of an aggregate of more than one volume.

Stacking is allowed only for the output data sets from a physical unload (UNLDDN) or a logical unload (OUTDDN). When you unload LOB or XML data, stacking output files on tape is not supported for CLOBF, DBCLOBF, or BLOBF columns.

TRTCH

Specifies the track recording method for magnetic tape drives that have improved data recording capability.

NONE

Specifies that a track recording method is not specified during dynamic allocation.

COMP

Specifies that data is written in compacted format.

NOCOMP

Specifies that data is written in standard format.

See *MVS JCL Reference* for more information about the PATHOPTS and PATHMODE options.

Related reference:

“User-allocated ddnames” on page 80

To run unload jobs, you must allocate certain ddnames in the DB2 HPU JCL.

“PROCESS OPTIONS block syntax and description” on page 83

Use the PROCESS OPTIONS control statement to change and restore the default ddnames that DB2 HPU uses for LISTDEF definition libraries and TEMPLATE definition libraries.

“How to use TEMPLATES”

As an alternative to using JCL to specify the data sets, you can use the TEMPLATE control statement to dynamically allocate DB2 HPU data sets.

How to use TEMPLATES

As an alternative to using JCL to specify the data sets, you can use the TEMPLATE control statement to dynamically allocate DB2 HPU data sets.

You can specify the following information with the TEMPLATE statement:

- The data set naming convention
- DFSMS parameters
- Disk allocation parameters

By using TEMPLATE statements, you can standardize data set allocation, which reduces the need to customize and alter DB2 HPU job streams.

You can specify a TEMPLATE statement in the SYSIN data set, preceding the UNLOAD command that references the TEMPLATE statement, or in a TEMPLATE library.

A TEMPLATE library is a data set that contains only TEMPLATE statements. You can specify a TEMPLATE data set DD name by using the TEMPLATEDD keyword

in the PROCESS OPTIONS statement. This specification applies to all subsequent UNLOAD commands until the end of input or until DB2 HPU encounters the next PROCESS_OPTIONS TEMPLATEDD(*ddname*) statement.

You can use a template name instead of a JCL-allocated ddname with the following keywords:

COPYDDN

Input image copy file.

DDLDDN

Input DDL specification for image copy files of dropped objects.

UNLDDN

Output file for physical unload.

OUTDDN

Output file for logical unload.

LOADDDN

Output file that contains the generated LOAD SYSIN for a logical unload.

Additionally, a template name can be used for unloading LOB columns by using LOB file reference CLOBF, DBCLOBF, or BLOBF data type attributes.

When a name is specified in the COPYDDN, DDLDDN, UNLDDN, OUTDDN, or LOADDDN keywords, DB2 HPU searches for the following items in the following sequence:

1. A DD statement that is allocated in the JCL that corresponds to the exact name specified in the SYSIN data set.
2. A template with the same name that is defined in the SYSIN data set.
3. A template with the same name that is defined in the TEMPLATE library.
4. A generic ddname (DDNxxxx) that is already allocated in the JCL that corresponds to the ddnames that are used for processing per partition. This item applies only to the COPYDDN, UNLDDN, and OUTDDN keywords.

To avoid allocation errors when an unload is processed in its initialization phase, DB2 HPU checks that all data set names that were generated from TEMPLATE statements are unique. If a duplicate statement is found, the INZT034E message is issued.

DB2 HPU always assumes that the object has at least 1000 records. Therefore, the allocated size for the output LOB files is at least 1000 times the size of the LOB. To get a better estimate of the size of the output data sets, specify YES for VUX036/RTSESTIM.

Related reference:

“TEMPLATE block syntax and description” on page 89

Use the TEMPLATE control statement to allocate data sets without using JCL DD statements when an UNLOAD command is processed.

Example: Unloading data from a table with an XML column by using a TEMPLATE statement

This example shows how to use a TEMPLATE statement to unload data from a table with an XML column.

In this example, data is unloaded from the HISTORY XML column in the DSN8910.CUSTOMER table into CLOBF files that are written in an output PDS.

The output PDS is named LABO.XMLFILES.P10459A. The other column in the SELECT statement is unloaded into a single data set that is allocated as ddname OUTPUT.

```
TEMPLATE XMLFILE DSN LABO.XMLFILES.P10459A(R&TMP.)
UNIT WORK DSNTYPE PDS DIR 20
  UNLOAD TABLESPACE
OPTIONS TEMPLATESET (TMP=:RECNUM)
QUIESCE YES
DB2 NO
SELECT CID,
  HISTORY
INTO      ,
  HISTORY_CLOBF VARCHAR(44) CLOBF XMLFILE
FROM DSN8910.CUSTOMER
FORMAT DSNTIAUL
OUTDDN(OUTPUT)
```

Related concepts:

“XML data processing” on page 68
DB2 HPU can unload XML data.

Example: Unloading data from a table with a LOB column by using a TEMPLATE statement

This example shows how to use a TEMPLATE statement to unload data from a table with a LOB column (COL_LOB).

In this example, data from the base table and data from the LOB column, which is unloaded as LOB files, are unloaded into HFS files.

```
TEMPLATE FILEL PATH '/u/test/MYDATA/LOB.&COL.&NUM.'
TEMPLATE FILED PATH '/u/test/MYDATA/BASE.&TS.'
TEMPLATE LOAD DSN MYLOAD.RELOAD(HPULOB1) UNIT WORK

UNLOAD TABLESPACE
DB2 NO

OPTIONS
TEMPLATESET (COL =: COLNAME,
             NUM =: RECNUM )

REFORMAT( TYPE CLOB INTO VARCHAR(100) CLOBF FILEL)

SELECT COL_INT, COL_LOB
FROM ME.MYTABLE
OUTDDN(FILED)
FORMAT DSNTIAUL
LOADDDN (LOAD) LOADOPT(RESUME NO REPLACE)
;
/*
```

After the statement is run, data is unloaded into the following HFS files. Data from the COL_INT column in the base table is unloaded into /u/test/MYDATA/BASE.MYTS. Data from the COL_LOB column is unloaded into the following files:

- /u/test/MYDATA/LOB.COL_LOB00000001
- /u/test/MYDATA/LOB.COL_LOB00000002
- /u/test/MYDATA/LOB.COL_LOB00000003
- ...

Each data set contains a single LOB.

Related concepts:

“LOB data processing” on page 68
 DB2 HPU can unload LOB data.

Example: Unloading into a data set allocated via a TEMPLATE with DSNTYPE EXTREQ

In this example, you unload data in XML format with an SQL statement involving SQL expressions from an indented table.

```

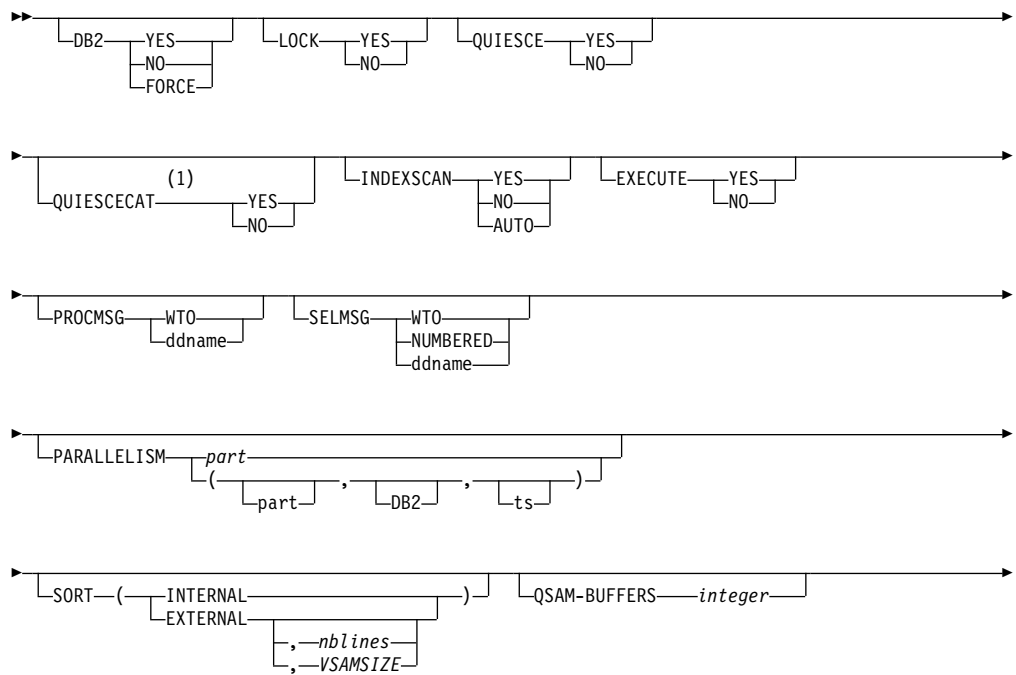
TEMPLATE XMLFILE DSN LABO.XMLFILES.P10459A(R&TMP.)
UNIT WORK DSNTYPE EXTREQ DIR 20
UNLOAD TABLESPACE
OPTIONS TEMPLATESET (TMP=:RECNUM)
QUIESCE YES
DB2 NO
SELECT CID,
HISTORY
INTO,
HISTORY_CLOBF VARCHAR(44) CLOBF XMLFILE
FROM DSN81010.CUSTOMER
FORMAT DSNTIAUL
OUTDDN(OUTPUT)
  
```

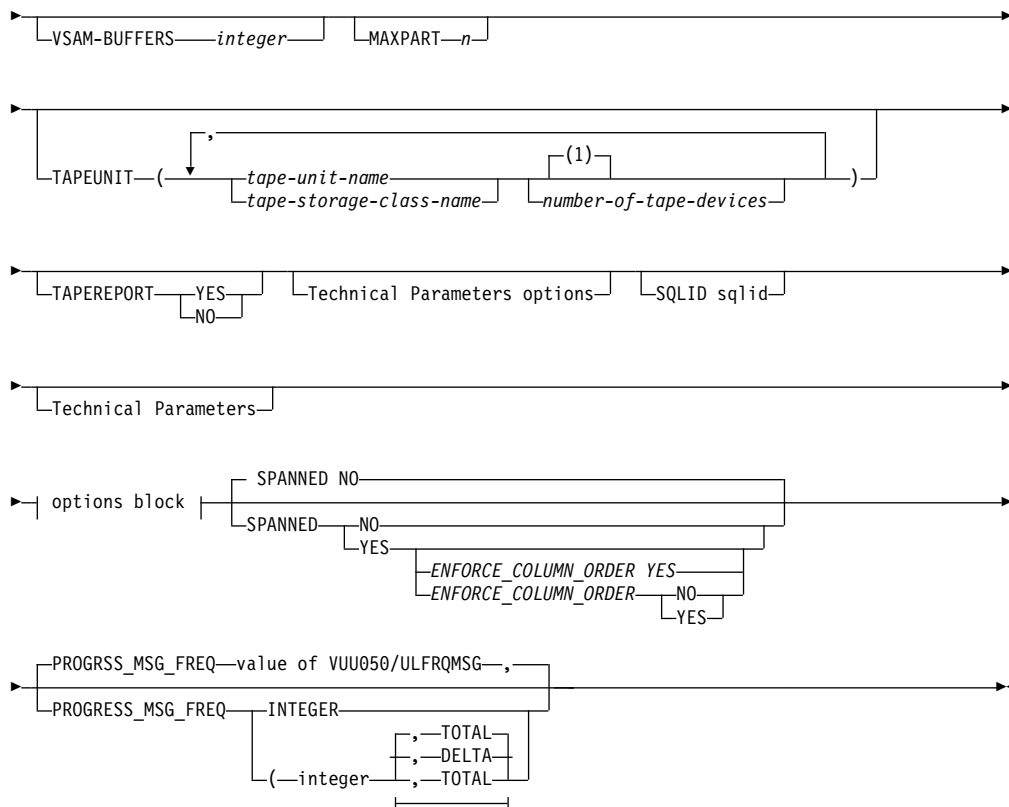
GLOBAL OPTIONS block syntax and description

Use the GLOBAL OPTIONS block to specify default values that apply to all UNLOAD blocks that you specify in your SYSIN. If you specify a keyword or value in both the GLOBAL OPTIONS block and the UNLOAD block, the value in the UNLOAD block is used.

A keyword that is specified in the GLOBAL OPTIONS block applies to all UNLOAD TABLESPACE statements. Keywords that are specified in an UNLOAD TABLESPACE statement apply only to the specific UNLOAD statement.

GLOBAL OPTIONS block





Notes:

- 1 The value of the QUIESCECAT keyword is forced to NO in DB2 10 for z/OS enabling-new-function mode and later releases.

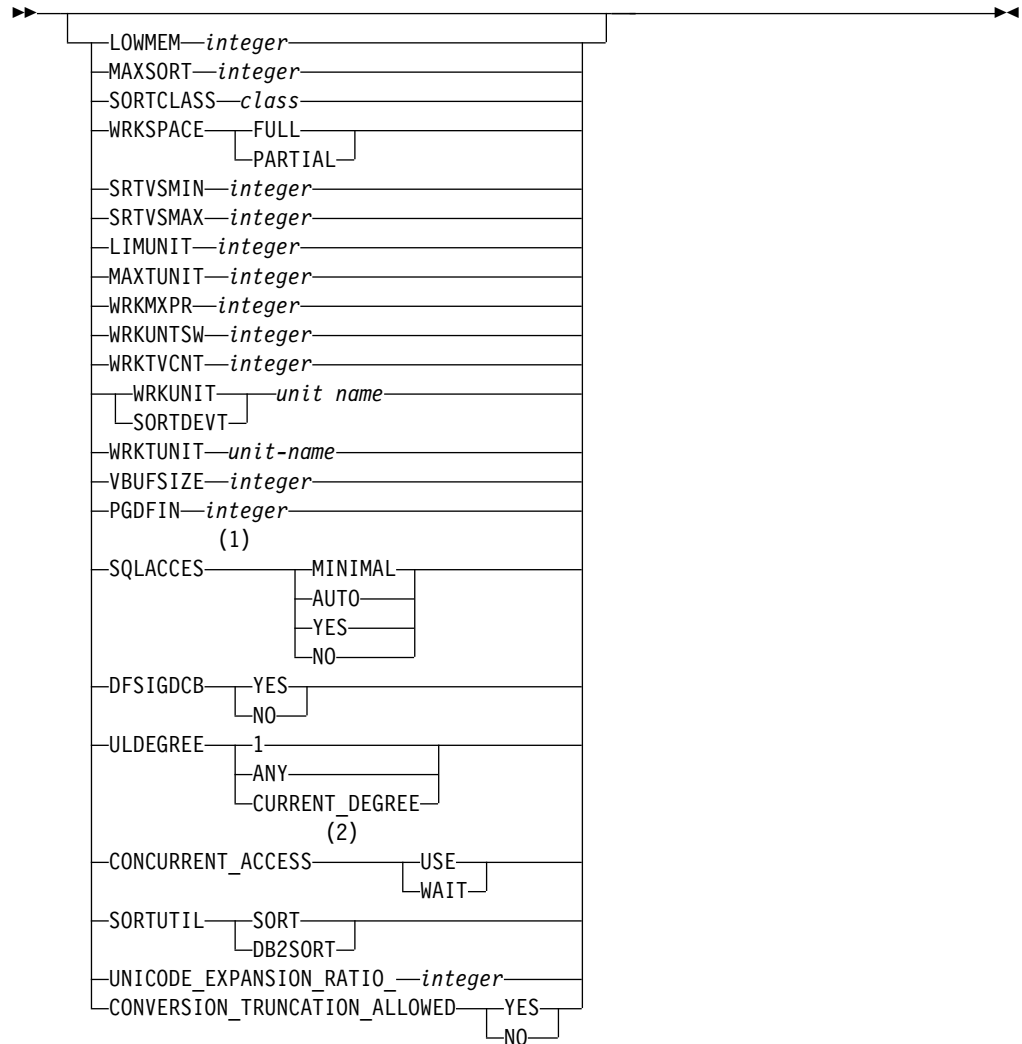
Technical Parameters options block

Use the Technical Parameters options block to override many of the processing parameters that are specified by the DB2 HPU settings. You can override these parameters in SYSIN to change a processing option for specific jobs without impacting other DB2 HPU jobs.

These keywords can be specified only in the GLOBAL OPTIONS block after the GLOBAL keyword. They apply to all the UNLOAD commands that are coded in the SYSIN.

See "Customization reference" on page 395 for descriptions of the corresponding PARMLIB parameters with the identical name. The descriptions give a detailed explanation of each keyword. The only exceptions are the SORTCLASS SYSIN keyword, which corresponds to the VUX020/SORTCLAS parameter, and the SORTDEVT keyword, which is an alias of WRKUNIT and corresponds to the VUM013/WRKUNIT parameter.

The same keyword can be specified only once.



Notes:

- 1 The value of the SQLACCES keyword is forced to YES in DB2 10 for z/OS enabling-new-function mode and later releases.
- 2 The corresponding PARMLIB parameter is VUU0061/ACCPREP.

Notes: UNICODE_EXPANSION_RATIO is associated to VUU068/ULUNIEXP parmlib parameter and CONVERSION_TRUNCATION_ALLOWED is associated to the VUU067/ULCNVTRC parmlib parameter.

Related reference:

“UNLOAD block syntax and description” on page 110
 See this topic for descriptions of the keywords in the GLOBAL OPTIONS block syntax.

Examples: Using committed data that is being updated or deleted

These examples show how you can use the CONCURRENT_ACCESS keyword when you unload data that is being updated or deleted. The data must be committed.

The following example shows how to specify the `CONCURRENT_ACCESS` keyword so that DB2 HPU uses committed data that is being updated or deleted.

```
GLOBAL CONCURRENT_ACCESS USE ;
UNLOAD TABLESPACE
QUIESCE YES
DB2 FORCE
SELECT * FROM DSN81010.EMP
FORMAT DSNTIAUL
OUTDDN(OUTPUT)
```

The following example shows how to specify the `CONCURRENT_ACCESS` keyword so that DB2 HPU does not use committed data that is being updated or deleted.

```
GLOBAL CONCURRENT_ACCESS WAIT ;
UNLOAD TABLESPACE
QUIESCE YES
DB2 FORCE
SELECT * FROM DSN81010.EMP
FORMAT DSNTIAUL
OUTDDN(OUTPUT)
```

Examples: Invoking sort utilities

These examples show how you can use the `SORTUTIL` keyword to invoke your on-site sort program or the DB2 Sort utility.

In the following example, the table space has 50 partitions. Several sort operations are required because of partition processing parallelism. Partition processing parallelism was requested because one output file is allocated per partition, and partition parallel processing is allowed because `PARALLELISM (5, ,)` is specified. The sort operations are run outside of DB2 because `DB2 NO` was specified. The `GLOBAL` statement specifies that DB2 Sort will do the sort operation.

```
GLOBAL SORTUTIL DB2SORT ;

UNLOAD TABLESPACE
PARALLELISM (5 , ,)
DB2 NO
DB2 NO
SELECT *
FROM MZFLB.TBDZ8AFLB20101
ORDER BY 1 DESC,
        2 DESC
OUTDDN(SYSREC)
FORMAT DSNTIAUL
;
```

In the following example, several sort operations are required because several `SELECT` statements with `ORDER BY` clauses are in a single `UNLOAD` command. The sort operations are run outside of DB2 because `DB2 NO` was specified. The `GLOBAL` statement specifies that DB2 Sort will do the sort operations.

```
GLOBAL SORTUTIL DB2SORT
      OPTIONS  TEMPLATESET(NUM=:SELNUM)
      ;
TEMPLATE SYSREC DSN MZFLB.&DB..S&NUM. UNIT WORK

UNLOAD TABLESPACE
DB2 NO
SELECT *
FROM MZFLB.TBDZ8AFLB00101
ORDER BY 1 DESC,
        2 DESC
OUTDDN(SYSREC)
```

```

SELECT C1, C2
FROM MZFLB.TBDZ8AFLB00101
ORDER BY 1 DESC,
        2 DESC
OUTDDN(SYSREC)
SELECT C1, C2, C3
FROM MZFLB.TBDZ8AFLB00101
ORDER BY 2 DESC,
        1 DESC
OUTDDN(SYSREC)
FORMAT DSNTIAUL
;

```

In the following example, several sort operations are required because several SELECT statements with ORDER BY clauses are in a single UNLOAD command. The sort operations are run outside of DB2 because SORT (EXTERNAL) is specified. Row selection is done in SQL mode because DB2 FORCE is specified. DB2 Sort will do the sort operations.

```

GLOBAL SORT(EXTERNAL)
    SORTUTIL DB2SORT
    OPTIONS  TEMPLATESET(NUM=:SELNUM)
;
TEMPLATE SYSREC DSN MZFLB.&DB..S&NUM. UNIT WORK
SPACE (100, 50) TRK
UNLOAD TABLESPACE
DB2 FORCE
SELECT *
FROM MZFLB.TBDZ8AFLB00101
ORDER BY 1 DESC,
        2 DESC
OUTDDN(SYSREC)
SELECT C1, C2
FROM MZFLB.TBDZ8AFLB00101
ORDER BY 1 DESC,
        2 DESC
OUTDDN(SYSREC)
SELECT C1, C2, C3
FROM MZFLB.TBDZ8AFLB00101
ORDER BY 2 DESC,
        1 DESC
OUTDDN(SYSREC)
FORMAT DSNTIAUL
;

```

In the following example, the DB2SORT value that was specified in the DB2 HPU settings for the VUX037/SORTUTIL parameter is ignored. The on-site sort program is used instead of DB2 Sort.

```

GLOBAL SORTUTIL SORT
    OPTIONS  TEMPLATESET(NUM=:SELNUM)
;
TEMPLATE SYSREC DSN MZFLB.&DB..S&NUM. UNIT WORK
UNLOAD TABLESPACE
DB2 NO
SELECT *
FROM MZFLB.TBDZ8AFLB00101
ORDER BY 1 DESC,
        2 DESC
OUTDDN(SYSREC)
SELECT C1, C2
FROM MZFLB.TBDZ8AFLB00101
ORDER BY 1 DESC,
        2 DESC
OUTDDN(SYSREC)

```

```

SELECT C1, C2, C3
FROM MZLFLB.TBDZ8AFLB00101
ORDER BY 2 DESC,
        1 DESC
OUTDDN(SYSREC)
FORMAT DSNTIAUL
;

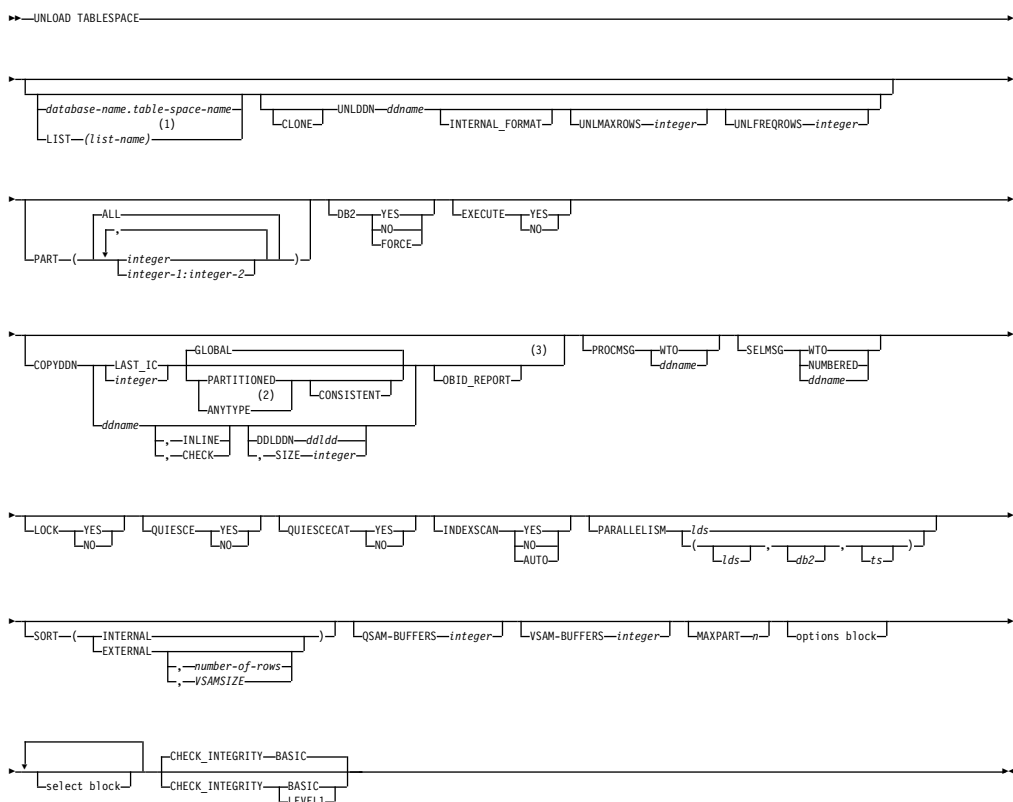
```

UNLOAD block syntax and description

See this topic for descriptions of the keywords in the GLOBAL OPTIONS block syntax.

The following diagram shows the syntax of the UNLOAD block:

UNLOAD block



Notes:

- 1 When you specify LIST (*list-name*) in the UNLOAD TABLESPACE command, you can do only a physical unload (using UNLDDN). You cannot specify a SELECT statement in the same UNLOAD TABLESPACE command. Additionally, you cannot do a physical unload with the same UNLOAD TABLESPACE command as a SELECT statement from a LIST (*list-name*) statement.
- 2 You can specify the ANYTYPE keyword only with COPYDDN LAST_IC.
- 3 COPYDDN -n with n > 1 and COPYDDN ddname and ANYTYPE are not supported if LOB or XML data is to be unload.

UNLOAD TABLESPACE

Identifies an UNLOAD statement. This UNLOAD statement is required.

database-bname.table-space-name

Specifies the complete name of the table space.

Restrictions:

- DB2 HPU does not support table spaces that are defined as work file.
- Do not use the following reserved words for the first word of *database-name.table-spacename* unless you delimit the first word with single or double quotation marks:
 - COPYDDN
 - DB2
 - LOCK
 - OPTIONS
 - PART
 - QUIESCE
 - SELECT
 - UNLDDN
 - CLONE
 - LIST
 - WITH

For example, the following command is incorrect:

```
UNLOAD TABLESPACE DB2.SELECT
```

The following forms of the command are correct:

```
UNLOAD TABLESPACE "DB2".SELECT
```

or

```
UNLOAD TABLESPACE 'DB2'.SELECT
```

LIST (*list-name*)

Identifies the name of a list of objects that are defined by a LISTDEF statement. The list can include table spaces, index spaces, databases, and partitions. The list cannot include LOB table spaces and directory objects. When you specify LIST (*list-name*), the referenced LISTDEF identifies the following objects:

- The table spaces from which the data is to be unloaded. You can use the LISTDEF pattern-matching feature.
- For partitioned table spaces, the partitions from which the data is to be unloaded. These partitions are defined by the INCLUDE, EXCLUDE, and PARTLEVEL keywords in the LISTDEF statement.

When you use the LIST keyword with a LISTDEF that represents multiple table spaces, you can define a data set TEMPLATE statement that corresponds to all the table spaces, and then specify the template name in the UNLDDN keyword. In this case, DB2 HPU associates a single table space with one output data set, except when partition parallelism is activated. To activate partition parallelism with a TEMPLATE statement, the data set name that is defined in the TEMPLATE statement must contain the &PART variable.

CLONE

Indicates that DB2 HPU is to do a physical unload of only the cloned tables from the specified table spaces when you unload a single table space by using the *database-name.table-space-name* variable or a list of table spaces by using a LISTDEF. Base tables in the specified table spaces are not unloaded. If the list contains table spaces that do not have a clone, these table spaces are not unloaded.

You can use the CLONE keyword only when you do a physical unload and specify the UNLDDN keyword.

UNLDDN *ddname*

Specifies that a physical unload of the table space is to be done, and specifies the *ddname* of the output data set.

The format of this data set is the same as the format when a DB2 REORG UNLOAD ONLY is done.

The *ddname* variable is the base *ddname* of the output data set.

To process partitioned table spaces in parallel, specify one *unldddnnn* statement for each partition (*unlddd01, unlddd02,... unldddnnn*), where *nnn* is a 1- to 7-digit sequential number that identifies a partition to be unloaded. During the unload process, data from each partition is directed to the corresponding *ddname*. If the corresponding *ddname* is allocated, it is used for the given partition. Otherwise, the base *ddname* is used if it is allocated. The following example uses one *unldddnnn* statement:

```
UNLOAD TABLESPACE DB1.TS1 PART(1,2,4,5) UNLDDN(MYDD)
```

If MYDD, MYDD01, and MYDD0004 are allocated, then MYDD contains the rows from partitions 2 and 5, MYDD01 contains the rows from partition 1, and MYDD0004 contains the rows from partition 4.

If you do not specify this keyword, specify OUTDDN on the SELECT statement.

Attention: When the UNLDDN keyword is used on a table space that contains a LOB column, the physical unload is performed only on the base table space, not on the LOB table space.

Important: When you do a physical unload, the UNLDDN keyword is valid only if a table space name or a list name is also specified in the UNLOAD TABLESPACE command. Although doing a physical unload with UNLDDN specified but without specifying the table space name was possible in previous releases of DB2 HPU, this deprecated syntax is no longer supported.

INTERNAL_FORMAT

Specifies that the physical unload will be made in DB2 INTERNAL format. This format is compatible with FORMAT INTERNAL in the DB2 LOAD utility.

Important: If the unloaded table space contains more than one table, the DB2 LOAD utility might not be able to reload the data into a table space with the same structure.

The following example shows how you can request a physical unload of data in the INTERNAL format. In this example, the entire DSN8D81A.DSN8S81E table space is unloaded.

```
UNLOAD TABLESPACE DSN8D81A.DSN8S81E
QUIESCE YES
DB2 NO
UNLDDN(OUTPUT)
INTERNAL_FORMAT
```

UNLMAXROWS *integer*

Specifies the maximum number of rows to unload for a physical unload. If

you are unloading a partitioned table space, which is processed on a partition-by-partition basis, the limit applies to each partition.

UNLFREQROWS *integer*

Specifies sampling frequency for a physical unload. One row of every *n* rows is written to the UNLDDN data set.

PART Specifies the partitions of the table space to be processed. Use the PART keyword only with partitioned table spaces. The PART keyword can be specified in the UNLOAD block and the SELECT block.

- If you do not specify the PART keyword in the UNLOAD block, ALL is the default value.
- If you do not specify the PART keyword in a SELECT block of this UNLOAD block, the PART specification in the UNLOAD block is used.
- If you specify the PART keyword in any SELECT statement, it overrides any values that are specified in the UNLOAD block.
- For each UNLOAD statement, the union of all the subsets of partitions that are selected in all SELECT blocks must be equal to the subset that is specified in the PART keyword unless you use UNLDDN.
- If a statement is processed by DB2, the PART keyword is ignored.

If you are unloading from partition-level full image copies, you must provide a unique name for the DD statement for each partition to be unloaded. For example, if you specify one SELECT statement to unload from partitions 1, 2, and 4 and one SELECT statement to unload from partitions 2 and 3, you must specify a DD statement with a unique name for each of the following image copy partitions: 1, 2, 3, and 4.

integer

Indicates which partitions are to be processed.

ALL Specifies that the entire table space is to be processed.

integer-1:integer-2

Designates a range of partitions from *integer-1* to *integer-2*. *Integer-1* must be positive and less than the highest partition number in the table space. *Integer-2* must be greater than *integer-1* and less than or equal to the highest partition number.

Important: When you use a list that was generated from a LISTDEF statement, use the PARTLEVEL keyword in the LISTDEF statement to select partitions.

The default value is ALL.

COPYDDN

Indicates that the source is an image copy.

ddname

Specifies the ddname that points to the image copy.

When you specify a ddname, no control is provided by using the information from the DB2 catalog on the image copy. The ddname must be an image copy data set. Specifying a FlashCopy[®] is not supported with *ddname* because the results are unpredictable. For DB2 Version 7, the image copy data set can be an incremental copy if the copy is not compressed. In DB2 Version 8 and later releases, DB2 HPU can unload from an image copy of a compressed table space if the image copy was created with the SYSTEMPAGES option. If the ddname points to an inline copy, such as a REORG

share level change, you must specify the `INLINE` option. If you do not specify `INLINE`, duplicate or missing lines can occur in the output, or the job can fail during the unload process itself. If the copy is inline or is not known, you must specify the `CHECK` option.

When you specify a `ddname`, DB2 HPU prevents allocation of VSAM data sets within the table space, and DB2 HPU must access an image copy except when the `SELECT` statement is issued by DB2.

When you specify a `ddname`, DB2 HPU works on archives, avoids possible contention, and improves performance because QSAM is more efficient than VSAM.

DB2 HPU can unload an image copy that contains the data in a table space whose structure does not match the table space that is described in the DB2 catalog or the table space that is described by the DDLDDN data set. These mismatched table space structures concern only the definition of partitioning, not the data itself (the table structure). The following table shows whether the mismatched table space structure is supported:

Table 18. DB2 HPU support for table space structures

Type of table space in the image copy	Nonpartitioned table space in the DB2 catalog or DDLDDN	Partitioned table space in the DB2 catalog or DDLDDN	Range-partitioned table space in the DB2 catalog or DDLDDN	Partition-by-growth table space in the DB2 catalog or DDLDDN
Nonpartitioned table space (simple or segmented)	Supported	Supported	Supported	Supported
Partitioned table space with a different number of partitions	Not supported	Supported	Supported	Supported
Range-partitioned table space with a different number of partitions	Not supported	Supported	Supported	Supported
Partition-by-growth table space	Not supported	Not supported	Not supported	Supported

Notes:

1. If you request a file per partition, informational message INZU394I is issued, and all rows are unloaded in the file that corresponds to partition 1.
2. Support is limited by the following restrictions:
 - Unloading from an image copy when the number of partitions in the image copy exceeds the number of partitions in the DB2 catalog.
 - Unloading into an output file per partition when the number of partitions in the image copy is less than the number of

partitions in the DB2 catalog. In this case, you can do only a global unload of all data into a single output file.

- Unloading LOB and XML can only be done from FlashCopy image copies and only the LAST_IC option is supported.

Unless you use a DDL input file by specifying DDLDDN, the image copy must correspond precisely to the online table space, particularly to the level of the object IDs (OBID) of DB2 objects. If you have issued a DROP command or a CREATE statement, you must specify the ORIGINOBID keyword in the SELECT statement.

To work with a simple table space with multiple data sets, concatenate these data sets under the same ddname.

You can specify a generic value for COPYDDN, as shown in the following example:

```
COPYDDN FIC*
```

Values are concatenated in the order that they are encountered in the JCL.

Attention: The ddname that results from the asterisk (*) is not checked. For example, if you specify COPYDDN ST*, all data sets that are allocated to the STEPLIB ddname are considered, and the job stops.

CHECK

Specifies that DB2 HPU determines whether the image copy is an inline copy. This option is not valid for generic ddnames.

INLINE

Allows you to specify that an image copy is of type **INLINE**, which means that this image copy was taken when DB2 HPU is run. If COPYDDN LAST_IC is specified, you do not need to specify **INLINE**.

As the **INLINE** copy processing requires the image copy to be sorted, do not specify **INLINE** for an image copy which is certainly not an inline image copy to avoid a useless sort operation to be performed.

DDLDDN *ddldd*

DB2 HPU can use an image copy as input for an unload even when the structure of the unloaded objects does not match the table definition in the DB2 catalog. This unload includes tables and table spaces that have been dropped since the creation of the image copy file and eventually re-created with a different structure, and image copies that were generated on another DB2 subsystem.

When you specify DDLDDN *ddldd*, DB2 HPU reads the definition of the DB2 objects that are in the specified file instead of the DB2 catalog. If you specify DDLDDN *ddldd*, the corresponding ddname or TEMPLATE statement must correspond to a file that contains the CREATE statements that correspond to the structure of the data in the unloaded image copy file. You can code DDLDDN only when the input image copy is allocated by using a ddname or a TEMPLATE statement. DDLDDN cannot be used with COPYDDN LAST_IC or COPYDDN *integer*.

If the input image copy is an inline copy, you must specify the `INLINE` option. The `CHECK` option of the `COPYDDN` keyword is not compatible with `DDLDDN`.

The `DDLDDN` file must contain the following `CREATE` statements in the following order:

1. `CREATE DATABASE`
2. `CREATE TABLESPACE`
3. `CREATE TABLE`
4. `CREATE INDEX` for the clustering index for an index partitioned table space

Any `ALTER` statement or `CREATE VIEW` statement in the `DDLDDN` file is ignored.

The DDL that is contained in the `DDLDDN` can specify an `OBID` in the `CREATE TABLE` statement. This `OBID` is the default for the `ORIGINOBID` option of the `SELECT` statement. If the DDL does not contain an `OBID`, the default value is 0 for the `ORIGINOBID` keyword on all `SELECT` statements for the `UNLOAD` command. For a multi-table table space, you must specify the `OBID` of each table by using `ORIGINOBID` either in the `CREATE TABLE` statements of the `DDLDDN` file or in the `SELECT` statements of the `SYSIN`.

SIZE *integer*

Specifies the number of rows contained in the image copy. It is used:

- to estimate the number of pages to sort when unloading data from an `INLINE` image copy. The number of pages to sort is calculated as: $(SIZE * \text{average length of rows}) / 1024 * VUX023 / PGDFIN$
- to specify the number of rows to sort when an `ORDER BY` clause is specified. For a partitioned tablespace, if data is unloaded by partition, the number of rows is divided by the number of partition.

Default value: the value of the `VUX003/SIZE` parameter

OBID_REPORT

Specifies that DB2 HPU generates in `SYSPRINT` a list of all `OBIDs` for rows of tables that are found in the input full image copy (`FIC`) when you unload from an image copy.

Attention: This option might have a negative impact on the overall performance of DB2 HPU when you unload from an image copy.

LAST_IC or *integer*

Indicates that the data to be unloaded must be retrieved from one image copy or a set of image copies determined by DB2 HPU by searching the DB2 catalog for image copies that fulfill the following conditions:

- The image copies are full image copies (`FIC`)
- The image copies are cataloged data sets
- The image copies are one of the following types:

- Regular or FlashCopy image copies if no LOB or XML data is to be unloaded
- FlashCopy if LOB or XML data are to be unloaded. Only LAST_IC is allowed.
- The set of image copies includes all the following necessary objects:
 - The table space that includes the table to be unloaded, if this table space is partitioned, the image copies set must include all the data of the involved partitions (see the GLOBAL/PARTITIONED/ANYTYPE options for more details). The later rule also applies to partitioned-by-size table spaces.
 - If the unload request involves an LOB or XML column, the base table space (see above considerations on partitioned objects that also apply) and both the auxiliary table space and the nodeid index for the LOB or XML column involved.
- All the images copies have the same START_RBA (see SYSIBM.SYSCOPY table) under the following conditions:
 - The CONSISTENT option is specified.
 - A LOB or XML column is to be unloaded. Only LAST_IC is allowed.

Specify LAST_IC to get the data unloaded from the last eligible set of image copies. As all the above conditions must be satisfied by the set of image copies, be aware that the image copy selected for a single object might not be the latest copy made for this object.

Specify a negative integer (-n) to request DB2 HPU to retrieve the data from the *n*th last set of eligible image copies. Specifying *n* greater than 1 is forbidden if LOB or XML columns are to be unloaded.

COPYDDN LAST_IC is a synonym of COPYDDN -1.

If the last image copy of a partitioned table space consists of one image copy per partition, the LAST_IC option is not supported unless you also specify the PARTITIONED keyword or the ANYTYPE keyword.

integer

Specify a negative integer (-*n*) to indicate that the unload must be processed from the *n*th last valid registered FIC. The LAST_IC is equivalent to -1.

You can specify the following keywords for COPYDDN LAST_IC or COPYDDN *integer*:

GLOBAL

This keyword is the default value when COPYDDN LAST_IC or COPYDDN *integer* is specified. If a nonpartitioned table space is processed, this value is ignored. However, for a partitioned table space, LAST_IC processing looks for a global FIC of the complete table space.

PARTITIONED

If the unloaded table space is not partitioned, the PARTITIONED keyword is ignored. However, for a

partitioned table space, LAST_IC processing looks only for FICs that were created for each partition. When you specify COPYDDN *n*, only full image copies per partition are counted.

When you specify the PARTITIONED keyword, the most recent FIC for all unloaded partitions is selected. If some of the unloaded partitions do not have an FIC, DB2 HPU issues an error message.

ANYTYPE

Allows COPYDDN LAST_IC processing to look for the most recent global FIC or FIC per partition. The ANYTYPE keyword can be specified only with COPYDDN LAST_IC.

If the ANYTYPE keyword is coded without partitions, DB2 HPU looks for the latest FIC for each partition and for the latest global FIC. If all FICs for each partition are more recent than the global FIC, the more recent FICs are used. Otherwise, the global FIC is used.

ANYTYPE cannot be used if a LOB or an XML column is to be unloaded.

CONSISTENT

Checks whether the selected FIC has the same START_RBA for all unloaded partitions. This option ensures consistency in the unloaded data. You can specify the CONSISTENT keyword with only the PARTITIONED keyword or the ANYTYPE keyword.

The following conditions affect the CONSISTENT keyword:

- If the selected FIC is a global FIC that was created for all the partitions in the table space, the CONSISTENT keyword is always verified.
- If the selected FIC is a copy per partition, the CONSISTENT keyword checks whether the START_RBA is identical to the selected input FIC of all unloaded partitions.
 - If the START_RBA is identical, the unload is processed.
 - If the START_RBA is different, depending on the PART keyword or the ANYTYPE keyword, DB2 HPU looks for the preceding FIC.

If you did not specify the CONSISTENT keyword, the START_RBA is not checked. You can check whether the available copies meet the specifications of the COPYDDN option by completing the following steps:

1. From the DB2 catalog, extract a list of the image copies for *database-name.table-space-name* and partition numbers that match the list that is specified by the PART keyword. If you specified the ANYTYPE keyword, you must add 0 to the list. Order this list by descending START_RBA.
2. Split the list into groups that have the same START_RBA value.

3. Depending on the option that you specified in the COPYDDN keyword, select the relevant group. If the LAST_IC keyword is specified, select the last group. If you specified an *integer*, select an *n*th older group.
4. Check whether the list of image copies covers the list of partitions to be unloaded (0 match the whole set of partitions).

Restriction: When an image copy is used, the LOCK and QUIESCE keywords are not supported.

DB2 Specifies the processing to be performed for SELECT statements that are not supported by DB2 HPU.

YES Indicates that if the SELECT statement is too complex to be handled directly by DB2 HPU, DB2 is called to extract the rows. A warning message is issued to report this occurrence. The overall return code is raised to 4.

NO Indicates that the SELECT statements must be processed by DB2 HPU. If a SELECT statement is not supported by DB2 HPU, an error occurs and processing stops. The control is done when the SYSIN is read and before an unload is processed.

FORCE

Indicates that DB2 must be used to extract the requested rows. Specifying DB2 FORCE is useful when the SELECT statement uses a filtering predicate that is efficiently processed through SQL, and the filtering factor is high. An informational message is issued in the report.

Restriction: When you use COPYDDN to unload data from an image copy, DB2 YES or DB2 FORCE is accepted if you specify COPYDDN_STRICT(NO) in the VUU030/ULOPTNS PARMLIB parameter. If you specify COPYDDN_STRICT(YES), unloading an image copy forces the default value of the DB2 option to NO, and specification of DB2 YES or DB2 FORCE in SYSIN is not supported.

When a logical unload is processed, DB2 HPU can retrieve the data to be unloaded by native mode or SQL mode. In native mode, DB2 HPU retrieves the data by a direct reading of VSAM linear data sets (LDS) of the table space. In SQL mode, DB2 HPU retrieves the data by passing the SELECT statement to DB2. For some complex SQL statements, which are called unsupported SELECTs, native mode is unavailable. To determine which access method to use for other statements, consider performance, output, and data consistency.

Performance

Native mode

Use native mode when you need to unload a large amount of table data or a partition of a table space.

SQL mode

Use SQL mode for SELECT statements for which DB2 access paths are efficient. For example, use SQL mode when you have a highly filtered WHERE clause with a predicate that is indexed, and access is index only.

Output

The output does not depend on the access mode. However, for the following cases, DB2 HPU does not create the same output in both modes. Consider these differences when you choose an access mode.

Table 19. Cases of different output in access modes

Output	Condition	Native mode	SQL mode
LOAD statement	The resulting table contains IDENTIFY columns.	The LOAD statement uses the IGNOREFIELDS option so that identity columns are correctly handled.	The LOAD statement describes the identity columns as standard, nonidentity columns.
Data format	FORMAT DSNTIAUL is specified where DSNTIAUL STRICT is not specified in the VUU030/ULOPTNS parameter.	String constants are unloaded as CHAR data.	String constants are unloaded as VARCHAR data.
	FORMAT USER is specified.	String constants are unloaded as CHAR data.	String constants are unloaded as VARCHAR data.
Output data	PART option is specified.	Data is unloaded normally.	The PART option is not supported. Use SQLPART instead.
	<ul style="list-style-type: none">COPYDDN and DB2 YES are specified with an unsupported SELECT statement.COPYDDN and DB2 FORCE are specified, but COPYDDN_STRICT was not specified in the VUU030/ULOPTNS parameter.	The unloaded data comes from the image copy.	The unloaded data comes from the table space.

Data consistency

Depending on the QUIESCE and LOCK keywords, which affect data consistency, the values of the unloaded data might not be the same in both access modes because data is not selected from the same sources, such as the table space LDS or data that is provided by DB2.

The access mode that DB2 HPU uses depends on whether the SQL is supported and on the DB2 option that you specify.

DB2 YES

Specify DB2 YES to ensure that DB2 HPU determines which access mode to use. DB2 HPU uses native mode when SQL statements are supported.

Consider performance when you specify DB2 YES because native mode is not always the fastest method.

When you apply maintenance and DB2 YES is specified, data might be processed in native mode instead of SQL mode because DB2 HPU might support SQL statements that were previously unsupported. Therefore, changes in the output might occur.

DB2 NO

Specify DB2 NO to ensure that DB2 HPU uses native mode.

DB2 FORCE

Specify DB2 FORCE to ensure that DB2 HPU uses SQL mode.

To select the best DB2 option in your environment, unload the data with DB2 NO specified, then unload the data with DB2 FORCE specified, and select the option that runs fastest. However, remember that when you specify DB2 FORCE, performance depends on the access paths that are chosen by DB2, the table, and the index statistics. Ensure that the output, such as the data format and the LOAD statement, from the selected mode meets your requirements. Consider specifying EXECUTE NO to check whether the SELECT statement is supported and the content of the related LOAD statement that was generated.

The default value is the value of the VUU011/ULSEDB2 PARMLIB parameter.

EXECUTE

Specifies whether the LOADDN file is generated without unloading the data.

YES DB2 HPU processes all LISTDEF, TEMPLATE, and UNLOAD commands that are specified in SYSIN, allocates the output files that correspond to the TEMPLATE definition, and unloads the data that is specified in the UNLOAD and SELECT statements. If you specify LOADDN, DB2 HPU generates the LOAD SYSIN.

NO DB2 HPU does not allocate the output files that correspond to the TEMPLATE definition except for the LOADDN files, and DB2 HPU does not unload the data that is specified in the UNLOAD and SELECT statements. However, DB2 HPU does process all LISTDEF, TEMPLATE, and UNLOAD commands that are specified in SYSIN. If requested, DB2 HPU writes the LOAD commands in the LOADDN.

Required FIC information that is used as input is displayed, and execution reports are generated, but DISPLAY and QUIESCE commands on the unloaded table spaces are not issued. If you specified LOCK YES in SYSIN, locks are not issued.

Important: When you use JCL-allocated output files instead of templates, the output files must be allocated because processing depends on the DD statements that are allocated in the JCL.

The default value is YES.

PROCMSG

Specifies where to write the information messages that correspond to the unload of the table space, partitions and indexes, or image copy file.

WTO Specifies that the messages from the unload process are issued as write-to-operator (WTO) messages in the system log.

ddname

Specifies that the messages from the unload process are issued in the corresponding ddname. If this ddname is not allocated in the JCL, it is dynamically allocated as SYSOUT=*. This ddname can be equal to SYSPRINT, in which case these messages might be mixed

with other DB2 HPU messages. All messages that are issued in the output file are prefixed with the system time in the *hh:mm:ss* format.

The default value is the value of the VUU051/PROCMSG PARMLIB parameter.

SELMSG

Specifies where to write the information messages INZX089 and INZX090, which are issued for each SELECT statement.

WTO Specifies that the messages that correspond to the SELECT statements are issued in WTO messages in the system log.

NUMBERED

Specifies that the messages that correspond to the SELECT statements are issued in a ddname for each SELECT statement. The ddname is dynamically allocated in SYSOUT=*. The format of the generated ddname is *UxxSyyyy*, where *xx* is the two-digit UNLOAD number and *yyyy* is the SELECT number for the unload. All messages that are issued in the output file are prefixed with the system time in the *hh:mm:ss* format.

ddname

Specifies that the messages that correspond to the SELECT statements are issued in the corresponding ddname. If this ddname is not allocated in the JCL, it is dynamically allocated as SYSOUT=*. This ddname can be equal to SYSPRINT, in which case these messages might be mixed with other DB2 HPU messages. All messages that are issued in the output file are prefixed with the system time in the *hh:mm:ss* format.

The default value is the value of the VUU052/SELMSG PARMLIB parameter.

LOCK Indicates whether DB2 HPU must lock the table during the unload.

YES The table is accessed in read-only mode while DB2 HPU runs.

NO DB2 HPU processes the table without changing its access mode. Specifying LOCK NO does not preclude DB2 from taking locks, if SQL Access is used to access the data.

QUIESCE

Specifies whether to issue a QUIESCE command against the table space before unloading it. If the unload is against an image copy and DB2 FORCE is specified, or if DB2 YES is specified and no SELECT statement is processed natively, the QUIESCE keyword is ignored.

YES The QUIESCE command is processed if the table space is not in COPY-pending status; otherwise, the table space is stopped and restarted.

NO The table space is processed without the QUIESCE command.

Attention: DB2 HPU operates on the physical VSAM data set level that is outside of DB2. If you run DB2 HPU on a table in which a new row was recently inserted, the unloaded data might not contain the new row. The unloaded data might not show the row because DB2 might not have externalized the data to DASD yet. This situation can occur when you use DB2 HPU without issuing a QUIESCE (or STOP) on the object. Be careful when you specify QUIESCE NO.

QUIESCECAT

Specifies whether to issue a QUIESCE command on the DB2 catalog table spaces before unloading data. If at least one unload requests QUIESCECAT, the QUIESCE is done only once before any catalog is accessed.

YES A QUIESCE is processed on the catalog tables.

NO A QUIESCE is not processed on the catalog tables.

QUIESCECAT is forced to NO in DB2 10 for z/OS enabling-new-function mode and later releases because DB2 HPU accesses the catalog in SQL only at that DB2 level.

INDEXSCAN

Specifies to unload the rows of a table space according to the order of its index cluster. When you specify the INDEXSCAN keyword, it is a substitute for the SORT utilities (DFSORT).

When you specify INDEXSCAN, all SELECT statements that are relative to the same table space are sorted according to the index cluster. If the pages of the index cluster are too unorganized, the INDEXSCAN keyword is deactivated, and DFSORT is run. See the JESMSGLG card of the job execution output reports for more information about the using the INDEXSCAN keyword.

The following conditions are required to use the INDEXSCAN keyword:

- The table space to be unloaded must have a valid index cluster.
- Only a table space can be unloaded.
- The table space to be unloaded must be single-table (partitioned or nonpartitioned).
- At least one SELECT statement of the unload must be native and must contain either an ORDER BY clause or an ORDER CLUSTER clause.

Restrictions: The INDEXSCAN keyword cannot be used in the following situations:

- Partition parallelism is activated, the table space is partitioned, and there is only one output file.
- The cluster is index is nonpartitioned, and the table space is partitioned.

NO The INDEXSCAN keyword is not activated.

AUTO

The following conditions are required to use the AUTO option:

- At least one native SELECT statement without a WHERE clause must be specified. However, depending on the order of the index cluster columns (complete or partial), it must have either an ORDER CLUSTER clause or an ORDER BY clause.
- There must be no SELECT statement with an ORDER BY clause that is not in the order of the index cluster columns (complete or partial).

YES The conditions for YES are the same as the conditions for AUTO except YES supports the WHERE clause.

The default value is AUTO.

PARALLELISM *lds* or (*lds,db2,ts*)

Specifies the parallelism degree for an unload. You can define parallelism for the following objects:

- Table spaces
- SELECT statements that are processed by DB2
- Partitions for a partitioned table space

lds Specifies the parallelism degree to use when a partitioned table space is unloaded, where *lds* indicates the maximum number of partitions that are processed in parallel.

When unloading a partitioned table space in a single output file, you can force partition parallelism by using the *lds* subparameter of the PARALLELISM keyword in the UNLOAD TABLESPACE command or by specifying YES for the VUU036/GBLPARAL PARMLIB parameter.

When a single output file is used as output and if GBLPARAL is set to NO and PARALLELISM *lds* or PARALLELISM(*lds,db2,ts*) is not specified in the UNLOAD command, DB2 HPU does not use partition parallelism. Valid values are greater than or equal to 1.

The default value is the value that is specified in the VUX025/PARALLEL PARMLIB parameter.

db2 Specifies the parallelism degree to use for SELECT statements that are processed by DB2 (using DB2 FORCE or DB2 YES with unsupported SELECT statements). This parallelism is defined in a single UNLOAD command. Parallelism is not possible for SELECT statements that are coded in separate UNLOAD commands. Valid values are greater than or equal to 1.

The default is the value that is specified in the VUX031/DB2PARAL PARMLIB parameter.

ts Specifies the maximum parallelism degree value to use when data is unloaded from several table spaces in the same UNLOAD command. This value applies when more than one supported SELECT statement that corresponds to different table spaces are coded in the same UNLOAD command. In this case, DB2 HPU uses the parallelism degree that is specified by the *ts* value to unload table spaces in parallel.

Attention: When at least one SELECT statement that is coded in the same UNLOAD command is processed by using DB2, the *ts* value is decreased by 1. For example, if you specify PARALLELISM(10,10,5), and if at least one unsupported SELECT statement is coded in the UNLOAD command, only four table spaces can be processed in parallel.

The default is the value that is specified in the VUX030/UTLPARAL PARMLIB parameter.

The variables *lds*, *db2*, and *ts* values are positional, and the commas are required even if you do not want to set the value for one subparameter. For example, to specify a parallelism degree of 10 for DB2-processed SELECT statements without changing other values, specify PARALLELISM(,10,).

For SELECT statements that are processed natively by DB2 HPU, the parallelism degrees that can be set by using the PARALLELISM keyword or by the corresponding PARMLIB parameters are limited by the value of

the VUX005/MAXSORT PARMLIB parameter when the ORDER CLUSTER clause or the ORDER BY clause is used.

SORT Specifies how to process the ORDER BY clause for a SELECT statement that is processed by DB2.

INTERNAL

Specifies that the ORDER BY clause is processed by DB2.

EXTERNAL

Specifies that the SELECT statement is processed by DB2 without the ORDER BY clause. DB2 HPU sorts the rows that are returned by DB2 to process the ORDER BY clause. DB2 HPU determines the estimated number of rows depending on which of the following options is specified:

number-of-lines

Specifies the estimated number of rows to be sorted.

VSAMSIZE

Specifies that DB2 HPU calculates the estimated number of rows to be sorted based on the size of the VSAM data sets.

If you specify EXTERNAL without specifying *number-of-lines* or **VSAMSIZE**, the value of the VUX003/SIZE PARMLIB parameter is equal to the number of sorted rows.

Restriction: The EXTERNAL keyword is supported only when the columns of the ORDER BY clause are part of the list of selected items. The **VSAMSIZE** option is accepted only for supported SELECT statements that are processed with DB2 FORCE.

The default value is INTERNAL.

QSAM-BUFFERS

Specifies the BUFNO parameter of the DCB for sequential QSAM access.

The default value is the value that is specified in the VUM022/QSBUFNO PARMLIB parameter.

MAXPART *n*

Specifies the maximum number of partitions that DB2 HPU can process in a single unload process. If the total number of partitions that need to be processed for a table space is larger than the value that you specify in **MAXPART *n***, DB2 HPU automatically splits the single unload into several unloads to respect the value of the **MAXPART** keyword. You can use this parameter to reduce the memory consumption of DB2 HPU. If the value of *n* is smaller than the value of the **PARALLELISM** keyword at the partition level, the effective partition parallelism is limited by the **MAXPART** keyword.

The **MAXPART** keyword affects only physical unloads and logical unloads that are processed natively by DB2 HPU.

The default value is the value of the VUU060/ULMAXPAR PARMLIB parameter. When you specify 0, no splitting is done. When you specify a nonzero value, ORDER BY and ORDER CLUSTER clauses can be used only when each partition is unloaded into a separate file. When you specify a nonzero value and a split is done, the **OUTMAXROWS** keyword or the **UNLMAXROWS** keyword applies to each partition.

Impact of the MAXPART parameter on the global parallelism

When you specify MAXPART, the parallelism between table spaces (the *ts* subparameter of the PARALLELISM keyword or the VUX030/UTLPARAL PARMLIB parameter) applies to each subset of partitions that are processed internally as a separate table space. Setting the MAXPART keyword impacts the global parallelism that is used when you unload a table space.

The following example shows how different values for the MAXPART keyword can affect a particular table space when the PARALLELISM keyword is specified:

Example: A table space with 1000 partitions and PARALLELISM(20, 5, 5)

- When you do not specify the MAXPART keyword or when you specify MAXPART 0, the complete table space with 1000 partitions is processed in a single set of partitions, and 20 partitions are unloaded in parallel. DB2 HPU allocates memory to correspond to the 1000 partitions.
- When you specify MAXPART 40, the 1000 partitions are split into 25 sets of 40 partitions. Five sets of 40 partitions are processed concurrently because parallelism between table spaces is set to 5. In each set of partitions, a parallelism degree of 20 is used between partitions. Memory usage corresponds to 200 (5 × 40) partitions, and the maximum global parallelism degree between partitions is 100 (5 × 20) partitions. If you specify MAXPART 40 and PARALLELISM(20, 5, 1), the memory usage corresponds to 40 partitions, and the global parallelism between partitions is 20.

Impact of the MAXPART keyword and the table space parallelism

Each set of partitions is processed as a separate table space so DB2 HPU starts to process a new set of partitions only when a previous set is completely processed. This type of processing can reduce the global partition parallelism under certain circumstances (for example, when sets that contain empty partitions are processed). To keep the global partition parallelism degree at its highest level, try specifying MAXPART 1, and change the parallelism between table spaces.

The following examples show how different settings for the MAXPART and PARALLELISM keywords can affect table space parallelism. These examples show that you can easily manage partition parallelism when you prioritize parallelism between table spaces rather than the MAXPART keyword.

Example: MAXPART 50, PARALLELISM(50, 5, 1)

These settings create sets of 50 partitions. One set of partitions is processed at a time. DB2 HPU starts to unload 50 partitions in parallel, but after one partition is unloaded, DB2 HPU does not start to unload a new partition because it must finish unloading the complete partition set before it starts to unload a new partition set.

Example: MAXPART 25, PARALLELISM(25, 5, 2)

These settings create of 25 partitions. Two sets of partitions

are processed in parallel. DB2 HPU starts to unload 50 (2 x 25) partitions in parallel. When one partition is unloaded, DB2 HPU waits for the set of 25 partitions to unload before it starts to unload a new partition because it needs to finish unloading the complete partition set before it starts to unload a new partition set.

Example: MAXPART 1, PARALLELISM(1, 5, 50)

These settings create sets that contain only one partition. DB2 HPU processes 50 sets of partitions in parallel. When one partition is unloaded, DB2 HPU starts to unload a new set.

TAPEUNIT

Specifies multiple tape units, tape storage classes, or both, and the number of associated devices that DB2 HPU uses when you request that output files be allocated on tape. DB2 HPU ensures that the maximum number of tape devices is not exceeded by automatically reducing the parallelism degree, if necessary.

When you use TAPEUNIT in a TEMPLATE statement that allocates output files on tape, ONDEMAND_RESOURCE_ALLOCATION(YES) is forced to allow multiple unloads to share a limited number of tape devices. A TEMPLATE statement allocates files on tape when the tape unit or tape storage class is defined by the TAPEUNIT parameter or when the template statement contains the tape option STACK or TRTCH. The VOLUMES and UNCNT parameters in the TEMPLATE statement are ignored, and scratch volumes are used.

When you unload to tape units, each physical unload or logical unload is serialized, and DB2 HPU reads the input files (table space or image copy files) once for each type of unload.

When OUTDDN or UNLDDN keywords specify several ddnames that allocate files on tape, each tape output file is generated by unloading the input data separately.

Using partition parallelism to unload a partitioned table space depends on the number of output files (one per partition or a single global unload file) and the number of available tape devices. When you request a global unload, a single tape device is used. Depending on the third value of the PARALLELISM keyword, parallelism can be used to read partitions. You can use MAXPART to limit the number of partitions that are read concurrently. When you request one output file per partition, MAXPART is forced to 1. Each partition can be unloaded on a separate tape device. The parallelism degree is defined by the third value of the PARALLELISM keyword. The maximum parallelism degree is limited by the number of usable tape devices that are defined by the TAPEUNIT keyword.

When you unload a partitioned table space into tape files, use either one global output file for all partitions, or use a separate file for each partition. You cannot unload some partitions in a global file and other partitions of the same table space into a file per partition.

When you use a TEMPLATE statement to allocate tape output files, the VOLUMES and UNCNT keywords in the TEMPLATE statement are ignored. In this case, scratch volumes are always used.

When you unload LOB or XML data, stacking output files on tape is not supported for CLOBF, DBCLOBF, or BLOBF.

tape-unit-name

Specifies the name of the tape unit.

tape-storage-class-name

Specifies the name of the tape storage class.

number-of-tape-devices

Specifies the maximum number of tape devices that DB2 HPU can use to allocate files on the specified tape unit or tape storage class.

Valid values are 1 - 4096.

The default value is 1 for tape units or storage classes that are specified in the TAPEUNIT keyword.

The following examples show how you can use TAPEUNIT.

Example: Using five tape devices to unload a set of table spaces that are based on the LISTDEF that is specified in SYSIN

One output file is created for each table that is found in the unloaded table spaces, and the files are stacked on the five tapes that are allocated with the unit name TAPE.

The parallelism degree at the table space level is set to five and is equal to the number of tape drives that are used for output files.

As the output files are stacked sequentially on tapes, if a table space contains several tables, the table space is read once for each table. Partitioned table spaces are read sequentially without partition parallelism (PARALLELISM(1,x,x)), and one single output file is created for each partitioned table space.

```
LISTDEF LSTTB INCLUDE TABLESPACES TABLE MZLFD.*
```

```
TEMPLATE FOUT DSN MZLFD.&LIST..&DB..&TS..&SEL.  
UNIT TAPE  
STACK YES
```

```
GLOBAL  
TAPEUNIT ( TAPE(5) )  
OPTIONS TEMPLATESET(SEL =: SELNUM)  
;
```

```
UNLOAD TABLESPACE  
QUIESCECAT NO QUIESCE NO  
DB2 NO
```

```
PARALLELISM(1,1,5)
```

```
SELECT * FROM LIST(LSTTB)  
OUTDDN(FOUT)  
FORMAT DSNTIAUL  
;
```

Example: Using 10 tape drives to unload 500 partitions of a partitioned table space

One output file is created for each partition, and the files are stacked on the 10 tapes that are allocated with unit name TAPE.

The MAXPART parameter is automatically forced to 1. The parallelism degree that is used to read the partitions is set by using the third subparameter of the PARALLELISM option (10).

However, if a larger value were used for parallelism in this example, the parallelism degree that is used to read the partitions would be limited by the number of available tape devices.


```

TEMPLATE FOUT DSN MZLFD.EX3.&DB..&TS..P&PART.
UNIT TAPE
STACK YES

GLOBAL
TAPEUNIT ( TAPE(10) )
;

UNLOAD TABLESPACE
QUIESCECAT NO QUIESCE NO
DB2 NO

PARALLELISM(1,1,10)

SELECT * FROM MZLFD.TBPART16;
OUTDDN(FOUT)
FORMAT DSNTIAUL

```

The TAPEUNIT keyword does not have default values.

TAPEREPORT

Specifies whether to generate a report about the tape usage. For each volume that was used, the report displays the data set names (DSN) and their positions on the tape (FILESEQ).

YES The tape usage report is generated.

NO The tape usage report is not generated.

If you do not specify TAPEREPORT, the tape usage report is generated according to the setting of the VZM011/TAPERPT PARMLIB parameter.

The default value is YES.

CHECK_INTEGRITY

Specifies the level of data integrity control to be performed when reading the input data from the LDS of a DB2 object from an Image Copy. It only applies to the native mode:

BASIC

Basic controls such as broken page indicator are performed, PGCOMB check..

LEVEL1

The same as basic control with additional control for LDS: check if the read page is the expected one.

Caution: higher levels (non-basic) of checking imply a performance decrease.

The default value is the value of VUX038/CHECKINT parmlib variable.

PROGRESS_MSG_FREQ

Specifies the frequency of the informational message that displays the current number of unloaded rows for each output file.

Integer

A non-null integer value means that this message is issued for each n rows. A null integer means that only one message is issued at the end of processing which indicates the total number of rows for each output file.

The default value is the value of the configuration parameter VUU050/ULFRQMSG

An additional parameter specifies interval time since the previous message was issued or since the first record was unloaded. Specify one of the following value:

DELTA

Messages INZX089 and INZX090 are issued to indicate the elapsed time since the previously issued message.

TOTAL

Messages INZX089 and INZX090 are issued to indicate the total elapsed time since the beginning of the unload process.

TOTAL is the default when a non-null value is specified and neither DELTA nor TOTAL is specified. TOTAL and DELTA are mutually exclusive.

The default value is TOTAL.

options block

See "OPTIONS block syntax and description" on page 134.

select block

See "SELECT block syntax and description" on page 168.

Related concepts:

"DB2 HPU-allocated ddnames" on page 79

DB2 HPU dynamically allocates some of the ddnames that are required to run unload jobs.

Related reference:

"User-allocated ddnames" on page 80

To run unload jobs, you must allocate certain ddnames in the DB2 HPU JCL.

"GLOBAL OPTIONS block syntax and description" on page 105

Use the GLOBAL OPTIONS block to specify default values that apply to all UNLOAD blocks that you specify in your SYSIN. If you specify a keyword or value in both the GLOBAL OPTIONS block and the UNLOAD block, the value in the UNLOAD block is used.

"Partition processing with LISTDEF and PARTLEVEL" on page 88

In DB2 HPU, a LISTDEF definition that includes the PARTLEVEL keyword without a partition number is supported only for compatibility purposes with the IBM LISTDEF utility and has no impact on how DB2 HPU unloads the data. LISTDEF definitions such as these do not activate partition parallelism.

Example: Unloading a table

In this example, you unload data from a table by using DB2 HPU.

The JCL in this example unloads data from the USER1.TBSCA table in the DBINFDM.TSSCA table space. Before the unload process begins, the table space is quiesced. All rows of the USER1.TBSCA table are unloaded in DSNTIAUL format into the USER1.UNLOAD01 file. A LOAD statement is generated in the SYSPUNCH data set.

```
//UNLOAD EXEC PGM=INZUTILB,PARM='DB81,DB2UNLOAD',
// REGION=0M,TIME=1440
//STEPLIB DD DSN=DB2UNL.SINZLINK,DISP=SHR
// DD DSN=DSN810.SDSNLOAD,DISP=SHR
//SYSREC00 DD DSN=USER1.UNLOAD01,
// DISP=(NEW,CATLG,DELETE),
// UNIT=SYSDA,
// SPACE=(CYL,(100,50),RLSE)
//UTPRINT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSPUNCH DD SYSOUT=*
```

```
//SYSIN DD *
UNLOAD TABLESPACE DBINFD.M.TSSCA
DB2 NO LOCK NO QUIESCE YES QUIESCECAT NO
SELECT *
FROM USER1.TBSCA
OUTDDN ( SYSREC00 )
FORMAT DSNTIAUL
LOADDDN SYSPUNCH
//
```

The JESMSGLG data set contains the following abbreviated information about the unloaded objects:

```
11.12.30 JOB01736 INZX006 TSSCA TABLESPACE UNLOAD PHASE STARTED
11.18.32 JOB01736 INZX090 SYSREC00 : 5002578 RECORDS WRITTEN
11.18.32 JOB01736 INZU222I SYSREC00 , TOTAL NUMBER OF RECORDS WRITTEN 5002578
```

The SYSPRINT data set contains the following information about the unloaded objects:

```
INZU224I IBM DB2 HIGH PERFORMANCE UNLOAD V4.1
INZI175I PROCESSING SYSIN AS EBCDIC.
-----1-----2-----3-----4-----5-----6-----7-----8
000001 UNLOAD TABLESPACE DBINFD.M.TSSCA
000002 DB2 NO LOCK NO QUIESCE YES QUIESCECAT NO
000003 SELECT * FROM TBSCA
000004 OUTDDN (SYSREC00 )
000005 FORMAT DSNTIAUL
000006 LOADDDN SYSPUNCH

DSNU000I DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = DB2UNLOAD
DSNU050I DSNUGUTC - QUIESCE TABLESPACE DBINFD.M.TSSCA
DSNU477I -DZ7M DSNUQUA - QUIESCE SUCCESSFUL FOR TABLESPACE DBINFD.M.TSSCA
DSNU474I -DZ7M DSNUQUA - QUIESCE AT RBA 0014C98DF148 AND AT LRSN 0014C98DF148
DSNU475I DSNUQUIB - QUIESCE UTILITY COMPLETE, ELAPSED TIME= 00:00:00
DSNU010I DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=0
INZU180I UTPRINT DD CARD IN JCL IS NOT USED WHEN VUX020/SORTCLAS IS SPECIFIED IN THE
PARMLIB OR WHEN SORTCLASS IS SPECIFIED IN SYSIN.
```

```
-----
INZU281I - UNLOAD STARTING AT POS(1, 1)
```

```
INZU277I - PROCESSING UNLOAD 00001 FROM TABLESPACE DBINFD.M.TSSCA
INZU279I - SELECT STATEMENTS USING SINGLE TABLE SPECIFICATION
INZU280I - SELECT 00001 STARTING AT POS(3, 1)
INZU282I OUTPUT DDNAME=SYSREC00
INZX090 SYSREC00 : 5002578 RECORDS WRITTEN
```

```
- TABLESPACE DBINFD.M.TSSCA - DB2 HIGH PERFORMANCE UNLOAD - STATISTICS - 10/20/06
* CREATOR.TABLE * OBID * ROWS READ * ROWS KEPT * TS PG READ *
*-----*-----*-----*-----*-----*-----*
* IBMUSER.TBSCA * 345 * 5002578 * 5002578 * *
*-----*-----*-----*-----*-----*-----*
* TOTAL UNLOAD STATISTICS ...* * 5002578 * 5002578 * 201103 *
* INVALID ROWS.....* * 0
* NUMBER OF PAGES IN ERROR...* * 0
```

```
INZU222I SYSREC00, TOTAL NUMBER OF RECORDS WRITTEN 5002578
INZU260I GENERATING LOAD STATEMENT FOR SELECT STARTING AT POS(3, 1)
```

The SYSPUNCH data set contains the following SYSIN data that can later be used to reload the objects:

```
LOAD DATA
LOG NO NOTIFY YES ENFORCE NO
INTO TABLE USER1.TBSCA
(
COLDATE POSITION( 1 )
DATE EXTERNAL( 10 ) , POSITION( 11 )
COLTIME POSITION( 11 )
TIME EXTERNAL( 8 ) , POSITION( 19 )
COLTIMESTAMP POSITION( 19 )
TIMESTAMP EXTERNAL( 26 ) ,
```

```

COLCHAR                POSITION( 45 )
CHAR( 8) ,
COLVARCHAR            POSITION( 53 )
VARCHAR                ,
COLLONGVARCHAR       POSITION( 130 )
VARCHAR                ,
COLINTEGER            POSITION( 4044 )
INTEGER                ,
COLSMALLINT           POSITION( 4048 )
SMALLINT              ,
COLFLOAT              POSITION( 4050 )
FLOAT( 53) ,
COLDECIMAL            POSITION( 4058: 4061 )
DECIMAL                ,
COLINTEGERNULL       POSITION( 4062 )
INTEGER
NULLIF( 4066) = '?' ,
COLCHARNULL          POSITION( 4067 )
CHAR( 8)
NULLIF( 4075) = '?'
)

```

Example: Unloading a table by using the last image copy

In this example, you unload data from a table by using the last image copy.

The JCL in the following example unloads data from the USER1.TBSCA table in the DBINFDM.TSSCA table space by using the last image copy.

```

//UNLOAD EXEC PGM=INZUTILB,PARM='DB81,DB2UNLOAD',
//          REGION=0M,TIME=1440
//STEPLIB DD DSN=DB2UNL.SINZLINK,DISP=SHR
//          DD DSN=DSN810.SDSNLOAD,DISP=SHR
//SYSREC00 DD DSN=USER1.UNLOAD01,
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=SYSDA,
//            SPACE=(CYL,(100,50),RLSE)
//UTPRINT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSPUNCH DD SYSOUT=*
//SYSIN DD *
UNLOAD TABLESPACE DBINFDM.TSSCA
COPYDDN LAST_IC
SELECT *
FROM USER1.TBSCA
OUTDDN ( SYSREC00 )
FORMAT DSNTIAUL
LOADDDN SYSPUNCH
//

```

The JESMSGLG data set contains the following abbreviated information about the unloaded objects:

```

13.28.14 JOB01349 INZX081 DSN8S10E IMAGE COPY IS BEING READ FROM DDNAME IC00
13.28.14 JOB01349 INZX062 W10001 SORT ACTIVATED VIA E15
13.28.15 JOB01349 INZX090 OUTPUT : 42 RECORDS WRITTEN IN 00:00:00, UNLOAD DONE

```

If the last image copy is a FlashCopy, the JESMSGLG data set contains the following abbreviated information about the unloaded objects:

```

13.28.14 JOB01349 INZX081 DSN8S10E FLASHCOPY IS BEING READ
13.28.14 JOB01349 INZX062 W10001 SORT ACTIVATED VIA E15
13.28.15 JOB01349 INZX090 OUTPUT : 42 RECORDS WRITTEN IN 00:00:00, UNLOAD DONE

```

The other parts of the report contain information like the information that is shown in “Example: Unloading a table” on page 130.

The following example shows how to unload a partitioned table space with five partitions that have copies per partition. Partitions 2 and 3 are standard copies, and partitions 1, 4, and 5 are FlashCopies.

```
UNLOAD TABLESPACE
PART (1,4:5)
QUIESCE YES
DB2 NO
COPYDDN LAST_IC
SELECT * FROM DSN81010.EMP
FORMAT DSNTIAUL
OUTDDN(OUTPUT) ;
UNLOAD TABLESPACE
PART (2:3)
QUIESCE YES
DB2 NO
COPYDDN LAST_IC
SELECT * FROM DSN81010.EMP
FORMAT DSNTIAUL
OUTDDN(OUTPUT)
```

Examples: Requesting progress messages via the UNLOAD command

Example 1: Request progression messages be issued every 100 rows. The messages will indicate the elapse time since the beginning of the unload.

```
PROGRESS_MSG_FREQ ; _____
UNLOAD TABLESPACE
DB2 NO
SELECT *
FROM DSN81010.EMP
FORMAT DSNTIAUL
OUTDDN (OUTPUT)
```

Example 2: Request progression messages be issued every 1000 rows. The messages will indicate the elapse time since the beginning of the unload.

```
GLOBAL
PROGRESS_MSG_FREQ (1000)
; _____
UNLOAD TABLESPACE
DB2 NO
SELECT *
FROM DSN81010.EMP
FORMAT DSNTIAUL
OUTDDN (OUTPUT)
```

Example 3: Request progression messages be issued every 1000 rows and that indicate the elapse time spent to unload for the latest set of 1000 rows.

```
GLOBAL
PROGRESS_MSG_FREQ (1000 , DELTA)
OPTION NULLPAD YES ; _____
UNLOAD TABLESPACE
DB2 NO
SELECT *
FROM DSN81010.EMP
FORMAT DSNTIAUL
OUTDDN (OUTPUT)
```

Example 4: Request progression messages be issued every 1000 rows and that indicate the elapse time spent from the beginning of the unload. TOTAL is not mandatory since this is the default value.

```
GLOBAL
PROGRESS_MSG_FREQ (1000 , TOTAL) ;_____
UNLOAD TABLESPACE
DB2 NO
SELECT *
FROM DSN81010.EMP
FORMAT DSNTIAUL
OUTDDN (OUTPUT)
```

Example: Requesting the input data integrity to be checked with the LEVEL1 controls

Requesting the input data integrity to be checked with the LEVEL1 controls, i.e DB2 HPU checks if the read page is the expected one.

```
UNLOAD TABLESPACE
DB2 NO
CHECK_INTEGRITY LEVEL1
SELECT *
FROM DSN81010.EMP
FORMAT DSNTIAUL
OUTDDN (OUTPUT)
```

OPTIONS block syntax and description

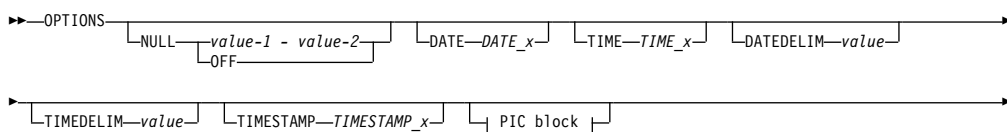
Use the OPTIONS block to specify the default conversions that are with the SELECT statements. This block can be used in the GLOBAL block, the UNLOAD block, and the SELECT block.

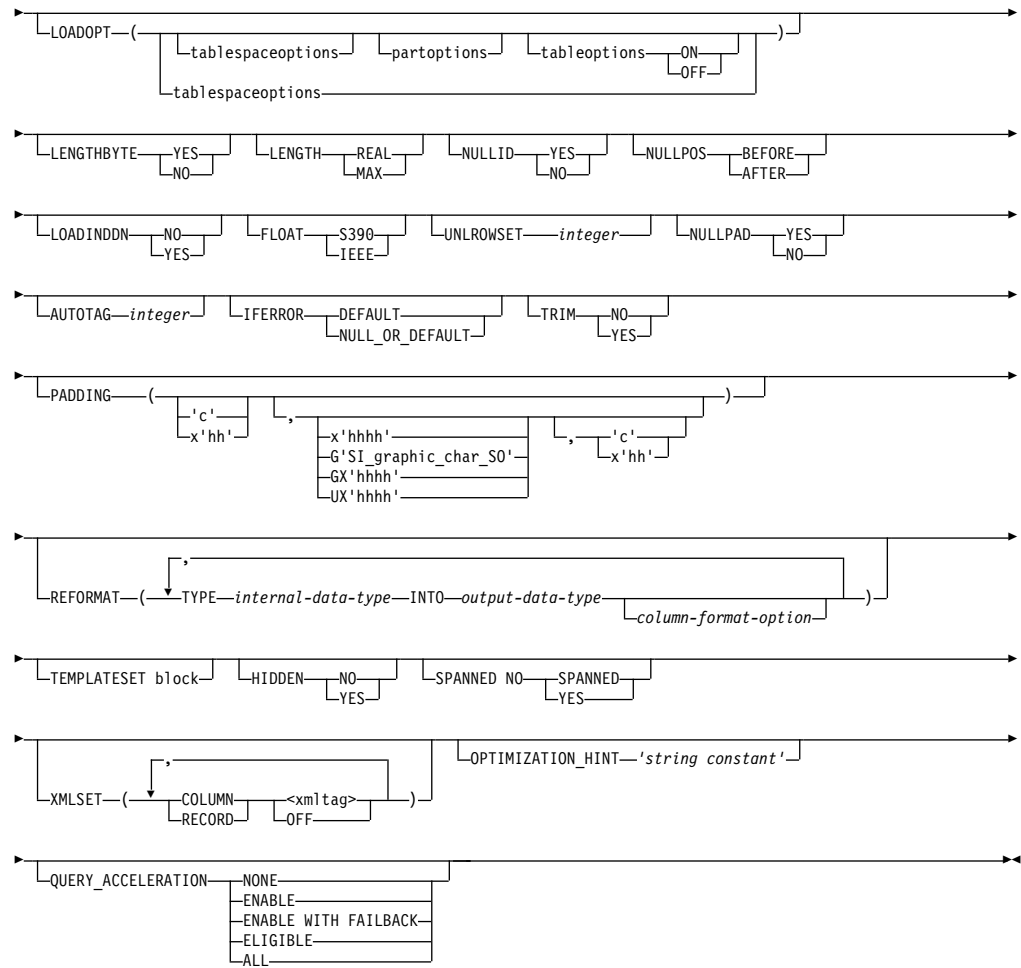
The effect of the options that are specified in the OPTIONS block at the GLOBAL level or at the UNLOAD level depends on the value that is specified for the VUU057/OPALLFMT PARMLIB parameter.

- If NO is specified for the VUU057/OPALLFMT parameter, all options that you specify in a GLOBAL block or an UNLOAD block apply only to the USER format except for the LOADOPT, FLOAT, UNLROWSET, LOADINDDN, AUTOTAG, and IFERROR options. The LOADOPT, UNLROWSET, LOADINDDN, AUTOTAG, and IFERROR options apply to all formats. The FLOAT option applies to DSNTIAUL, USER, and VARIABLE formats.
- If YES is specified for the VUU057/OPALLFMT parameter, all options that you specify in an OPTIONS block at the GLOBAL level or the UNLOAD level apply to all formats except for some specific options that have no meaning for some formats and for LENGTHBYTE and LENGTH options that are specific to the USER format.

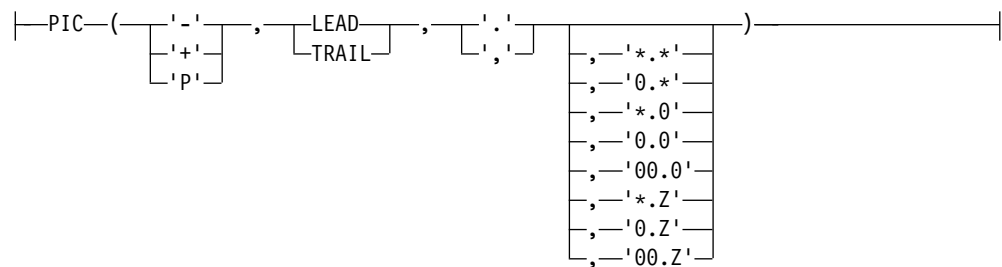
The following diagram shows the syntax of the OPTIONS block:

OPTIONS block





PIC block:



The following table shows the default values for options that are specified in the OPTIONS block. Each option can have a different default value based on the output format.

Table 20. Default values per output format for options that are specified in the OPTIONS block

Option name	Applies only to FORMAT USER if OPALLFMT=NO (GLOBAL or UNLOAD levels)	Applies to all formats if OPALLFMT=YES	PARMLIB parameter	Default value for FORMAT DELIMITED ¹	Default value for FORMAT DSNTIAUL ¹	Default value for FORMAT EXTERNAL ¹	Default value for FORMAT USER ²	Default value for FORMAT VARIABLE ³
NULL	Yes	Except FORMAT DELIMITED	VUU014/ULNULL	OFF	X'6F X'00' or VUU014/ULNULL	X'6F X'00' or VUU014/ULNULL	VUU014/ULNULL	X'FF X'00' or VUU014/ULNULL
NULLLID	Yes	Yes	None	NO	YES	YES	YES	YES
NULLPOS	Yes	Except FORMAT DELIMITED	VUU022/NULLPOS	N/A	AFTER or VUU022/NULLPOS	AFTER or VUU022/NULLPOS	VUU022/NULLPOS	BEFORE or VUU022/NULLPOS

Table 20. Default values per output format for options that are specified in the OPTIONS block (continued)

Option name	Applies only to FORMAT USER if OPALLFMT=NO (GLOBAL or UNLOAD levels)	Applies to all formats if OPALLFMT=YES	PARMLIB parameter	Default value for FORMAT DELIMITED ¹	Default value for FORMAT DSNTIAUL ¹	Default value for FORMAT EXTERNAL ¹	Default value for FORMAT USER ²	Default value for FORMAT VARIABLE ¹
DATE	Yes	Yes	VUU015/ULDATE	DATE_C or VUU015/ULDATE	DATE_DB2 or VUU015/ULDATE	DATE_DB2 or VUU015/ULDATE	VUU015/ULDATE	DATE_C or VUU015/ULDATE
DATEDELIM	Yes	Yes	VUU055/DATEDEL	'.' or VUU055/DATEDEL	Delimiter of the default DB2 format or VUU055/DATEDEL	Delimiter of the default DB2 format or VUU055/DATEDEL	VUU055/DATEDEL	'.' or VUU055/DATEDEL
TIME	Yes	Yes	VUU016/ULTIME	TIME_A or VUU016/ULTIME	TIME_DB2 or VUU016/ULTIME	TIME_DB2 or VUU016/ULTIME	VUU016/ULTIME	TIME_A or VUU016/ULTIME
TIMDELIM	Yes	Yes	VUU056/TIMEDEL	'.' or VUU056/TIMEDEL	Delimiter of the default DB2 format or VUU056/TIMEDEL	Delimiter of the default DB2 format or VUU056/TIMEDEL	VUU056/TIMEDEL	'.' or VUU056/TIMEDEL
TIMESTAMP	Yes	Yes	VUU017/ULTMSTP	TIMESTAMP_B or VUU017/ULTMSTP	TIMESTAMP_B or VUU017/ULTMSTP	TIMESTAMP_B or VUU017/ULTMSTP	VUU017/ULTMSTP	TIMESTAMP_B or VUU017/ULTMSTP
PIC	Yes	Yes	VUU018/ULPIC	-LEAD. or VUU018/ULPIC	-LEAD. or VUU018/ULPIC	-LEAD. or VUU018/ULPIC	VUU018/ULPIC	-LEAD. or VUU018/ULPIC
Mask part of the PIC option	Yes	Yes	VUU032/ULMASK	0.0 or VUU032/ULMASK	0.0 or VUU032/ULMASK ³	0.Z or VUU032/ULMASK	VUU032/ULMASK	0.0 or VUU032/ULMASK ³
LOADOPT	No	N/A	N/A	VUU019/UOPTLDT and VUU020/UOPTLDP	VUU019/UOPTLDT and VUU020/UOPTLDP	VUU019/UOPTLDT and VUU020/UOPTLDP	VUU019/UOPTLDT and VUU020/UOPTLDP	VUU019/UOPTLDT and VUU020/UOPTLDP
LENGTHBYTE	Yes	No	None	NO	YES	YES	YES	YES
LENGTH	Yes	No	None	REAL	MAX	MAX	Depends on LENGTHBYTE	REAL
LOADINDDN	No	N/A	N/A	NO or contents of VUU030/UOPTNS	NO or contents of VUU030/UOPTNS	NO or contents of VUU030/UOPTNS	NO or contents of VUU030/UOPTNS	NO or contents of VUU030/UOPTNS
FLOAT	No	N/A	None	S390	S390	S390	S390	S390
UNLROWSET	No	N/A	N/A	VUU035/ULROWSET	VUU035/ULROWSET	VUU035/ULROWSET	VUU035/ULROWSET	VUU035/ULROWSET
NULLPAD	Yes	Yes	None	YES	NO	YES	YES	YES
PADDING	Yes	Yes	None	Space	Space for characters or graphic strings or x'00' for other data types	x'00'	Space	Space
AUTOTAG	No	N/A	N/A	VUU046/ULAUTAG	VUU046/ULAUTAG	VUU046/ULAUTAG	VUU046/ULAUTAG	VUU046/ULAUTAG
IFERROR	No	N/A	None	None	None	None	None	None
TRIM	Yes	Yes	VUU059/ULTRIM	NO or VUU059/ULTRIM	NO or VUU059/ULTRIM	NO or VUU059/ULTRIM	VUU059/ULTRIM	NO or VUU059/ULTRIM
SPANNED	NO	N/A	N/A	NO	NO	NO	NO	NO
ENFORCE_COLUMN_ORDER		N/A	N/A	YES	YES	YES	YES	YES

Notes:

1. When a value or a parameter name is specified in the table, you can define a default value for each format in the corresponding PARMLIB parameter. If the default value is not defined in the PARMLIB for the format, the hard-coded default value applies.
2. When a variable is specified for the USER format, the default value is the content of the corresponding PARMLIB parameter unless a hard-coded default value is specified in the table.
3. The default value applies to the DSNTIAUL and VARIABLE formats when a LIKE keyword that forces conversion to a CHAR data type is used or when an INTO clause or REFORMAT clause is used.

NULL Indicates whether the null indicator is generated in the output data set. You can specify NULL in the SELECT statement in the FORMAT USER syntax also.

The following values can be specified for NULL. Although *value-1* and *value-2* can be specified as 1 - 4 bytes or characters, you must specify both

value-1 and *value-2* as the same number of bytes or characters. You can specify *value-1* and *value-2* in character ('c') or hexadecimal (X'hh') formats.

value-1

The value of the null indicator when the column value is NULL.

value-2

The value of the null indicator when the column value is NOT NULL.

OFF A null indicator is not generated.

When FORMAT DELIMITED is specified, the default value is OFF. You cannot change this value in the OPTIONS block even when it is specified at the SELECT level.

The default values for other formats are described in "OPTIONS block syntax and description" on page 134.

DATE *DATE_x*

Specifies the default output format for the DATE columns, where *x* is any uppercase alphabetic character A - R or an integer 0 - 7.

When a date column is converted into the CHAR format by specifying FORMAT USER or by using either a REFORMAT clause or an INTO clause, the DATE format that is specified in the OPTIONS block is not used. Instead, the default date format of your environment is used.

The default values for formats are shown in "OPTIONS block syntax and description" on page 134.

You can override the DATE option at the column level with the TYPE keyword in a SELECT statement for the USER format, an INTO clause, or a REFORMAT clause.

DATEDELIM *value*

Specifies the default delimiter that is used in external date representations, where *value* must be one character and must be 1 byte long, regardless of the literal CCSID.

The default delimiter that is used when DATE columns are unloaded depends on the output format that is used, as shown in "OPTIONS block syntax and description" on page 134

TIME *TIME_x*

Specifies the default conversion for time representations, where *x* is any uppercase alphabetic character A - E or the integer 0.

When a time column is converted into a CHAR data type by specifying FORMAT USER or by using either a REFORMAT clause or an INTO clause, the time format that is specified in the OPTIONS block is not used. Instead, the default time format of your environment is used.

The default values for formats are shown in "OPTIONS block syntax and description" on page 134.

You can override the TIME option at the column level by specifying the TYPE keyword in a SELECT statement for the USER format, an INTO clause, or a REFORMAT clause.

TIMEDELIM *value*

Indicates the default delimiter that is used in external time representations, where *value* must be one character and must be 1 byte long, regardless of the literal CCSID.

The default values for other formats are described in “OPTIONS block syntax and description” on page 134.

TIMESTAMP *TIMESTAMP_x*

Specifies the default conversion for the **TIMESTAMP** columns, where *x* is an uppercase alphabetic character A - G or the integer 0.

When a timestamp column is converted into a **CHAR** data type by specifying **FORMAT USER** or by using either a **REFORMAT** clause or an **INTO** clause, the timestamp format that is specified in the **OPTIONS** block is not used. Instead, the DB2 standard timestamp external format is used.

The default values for formats are shown in “OPTIONS block syntax and description” on page 134.

PIC Defines the numeric data display format that is used when numeric values are converted for external representation. The **PIC** keyword has four parameters. You must specify the first three parameters. The fourth parameter is optional.

- The first parameter specifies the rules for printing the sign.
 - Indicates that the minus character (-) is present if the number is negative. Otherwise, the sign character is a blank.
 - + Indicates that the sign is always present. Positive values have a plus character (+), and negative values have a minus character (-).
 - P** Indicates that the padding character is used for positive values and the minus character (-) is used for negative values.
- The second parameter specifies the position the sign relative to the column.

LEAD The sign is placed in front of the numeric value. **LEAD** is ignored for floating point numbers.

TRAIL

The sign is placed after the numeric value. **TRAIL** is ignored for floating point numbers.

- The third parameter specifies the decimal separator.
 - .
 - ,
- The optional fourth parameter indicates the formatting rules that are used to display nonsignificant zeros for **DECIMAL**, **SMALLINT**, and **INTEGER** values.

The following table lists the valid values and explains their impact.

- The **Left padding** column indicates whether the number is padded on the left with nonsignificant zeros to fill the output field. The sign character is added to the left of these zeros.
- The **Left zero** column indicates whether a zero is displayed on the left of the decimal separator when the integer part of the number is 0.
- The **Decimal separator** column indicates whether the decimal separator is displayed.
- The **Right padding** column indicates whether the number is padded on the right with zeros up to the number of digits of the decimal scale.

Notes:

- The Decimal separator column applies only to decimal data. The decimal separator is never displayed for SMALLINT or INTEGER values.
- The ULMASK variable has no impact on the formatting of values for FLOAT.

Table 21. Nonsignificant zero values for DECIMAL and SMALLINT/INTEGER

ULMASK value	Left padding	Left zero	Decimal separator	Right padding
.*	No	Only when the value is 0	Only if the decimal value is not 0	No
0.*	No	Always	Only if the decimal value is not 0	No
*.0	No	Only when the decimal scale is 0	Only if the decimal scale is not 0	Yes
0.0	No	Always	Only if the decimal scale is not 0	Yes
00.0	Yes	Always	Only if the decimal scale is not 0	Yes
*.Z	No	Only when the decimal scale is 0	Always	Yes
0.Z	No	Always	Always	Yes
00.Z	Yes	Always	Always	Yes

The following table contains examples of the effect of the mask parameter when the following decimal values are converted:
DEC(5,2) converted to a CHAR(8)

Table 22. Mask parameter effects when the decimal values 5,2 are converted

Mask/value	001.40	000.40	123.00	-123.45	000.00
.*	1.4	.4	123	-123.45	0
0.*	1.4	0.4	123	-123.45	0
*.0	1.40	.40	123.00	-123.45	.00
0.0	1.40	0.40	123.00	-123.45	0.00
00.0	0001.40	0000.40	0123.00	-0123.45	0000.00
*.Z	1.40	.40	123.00	-123.45	.00
0.Z	1.40	0.40	123.00	-123.45	0.00
00.Z	0001.40	0000.40	0123.00	-0123.45	0000.00

The following table contains examples of the effect of the mask parameter when decimal values are converted:
DEC(5,0) converted to a CHAR(8)

Table 23. Mask parameter effects when the decimal values 5,0 are converted

Mask/value	12345.	00001.	-12345.	00000.
.	12345	1	-12345	0
0.*	12345	1	-12345	0
*.0	12345	1	-12345	0
0.0	12345	1	-12345	0
00.0	0012345	0000001	-0012345	0000000
*.Z	12345.	1.	-12345.	0.
0.Z	12345.	1.	-12345.	0.
00.Z	012345.	000001.	-012345.	000000.

The following table contains examples of the effect on the mask parameter when decimal values are converted:

DEC(5,5) converted to a CHAR(9)

Table 24. Mask parameter effects when the decimal values 5,5 are converted

Mask/value	.12345	.10000	-.12345	.00000
.	.12345	.1	-.12345	0
0.*	0.12345	0.1	-0.12345	0
*.0	.12345	.10000	-.12345	.00000
0.0	0.12345	0.10000	-0.12345	0.00000
00.0	00.12345	00.10000	-00.12345	00.00000
*.Z	.12345	.10000	-.12345	.00000
0.Z	0.12345	0.10000	-0.12345	0.00000
00.Z	00.12345	00.10000	-00.12345	00.00000

The default value of the PIC keyword that is used depends on the output format that is used, as shown in "OPTIONS block syntax and description" on page 134.

LOADOPT

Modifies the options of the DB2 LOAD command. Specify the options that you want DB2 HPU to place in the LOAD SYSIN that is created during the unload process.

The value of the LOADOPT keyword is created by merging values that are specified in the PARMLIB, the GLOBAL block, the UNLOAD block, and the SELECT block. However, if LOADOPT is also specified in the FORMAT specification, the LOADOPT keyword is used as is. It is not merged with previous levels.

tableoptions

Options for the table

partoptions

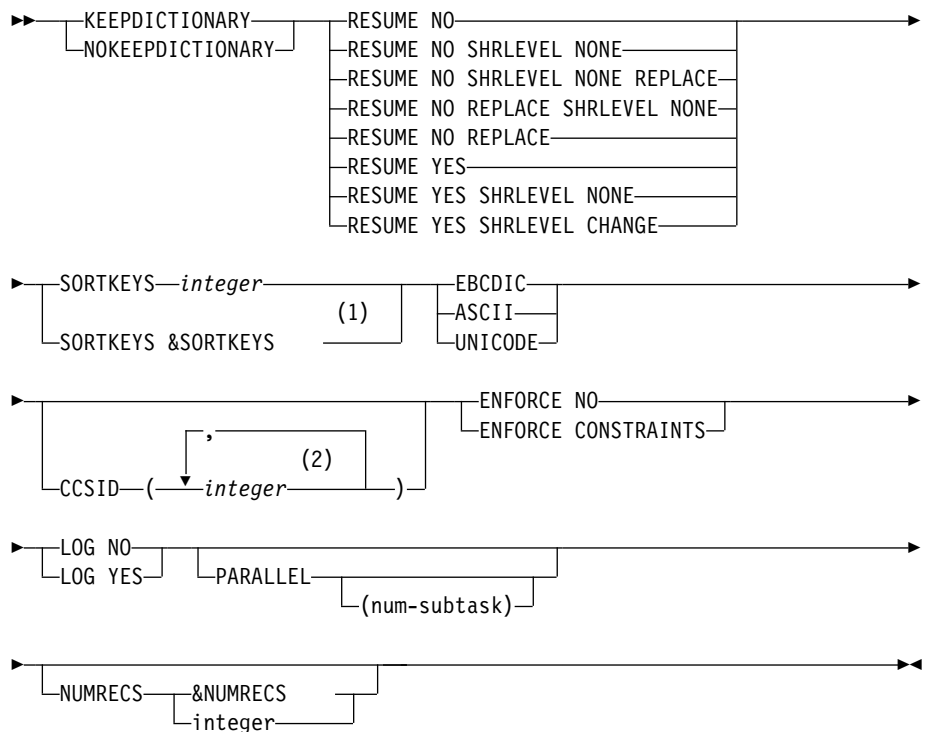
Options for the partition

tablespaceoptions

Options for the table space

The following syntax diagram shows the LOADOPT keywords that are recognized by DB2 HPU:

loadopt keywords



Notes:

- 1 &SORTKEYS is replaced with the actual value after data is unloaded except when EXECUTE NO is specified. When EXECUTE NO is specified, &SORTKEYS is replaced with 0.
- 2 Repeat up to three times.

The load options in the PARMLIB and any OPTIONS blocks that you specify in the GLOBAL block or the UNLOAD block are merged. If you specify the same option in two or more places, they are merged with the following priority:

1. UNLOAD block
2. GLOBAL block
3. PARMLIB

Any keywords that you specify that are not shown in the LOADOPT syntax diagram are added to the total option sequence. They are not merged with other options.

Important: If a LOADOPT keyword is specified in the FORMAT block, it is used as is, and it is not merged.

Note: If the DB2 HPU spanned YES option is specified, do not specify the FORMAT keyword in the LOADOPT option as this can lead to a wrong LOAD statement notified at LOAD time by either a syntax error message or by an error during the load processing.

LENGTHBYTE

Specifies whether the two length bytes for variable-length columns are

written to the output data set. LENGTHBYTE applies only when FORMAT USER is specified. It is ignored when other output formats are specified.

YES Writes the two length bytes.

NO Does not write the two length bytes.

The default value is YES when FORMAT USER is specified. You can change the default value by specifying an OPTIONS block at the GLOBAL or SELECT level or by changing LENGTHBYTE for a specific column in the FORMAT USER definition.

The following rules apply when you specify FORMAT USER:

- When LENGTHBYTE YES is specified, the default value for the LENGTH keyword is REAL.
- When LENGTHBYTE NO is specified, the default value for the LENGTH keyword is MAX.
- When LENGTHBYTE NO and LENGTH REAL are specified, DB2 HPU cannot generate a LOADDDN and issues the INZU203E error message if a LOADDDN file was requested.

When you specify FORMAT DELIMITED, the default value is NO. You cannot change this value by specifying an OPTIONS block, even when it is specified at the SELECT level.

When you specify FORMAT DSNTIAUL, FORMAT EXTERNAL, or FORMAT VARIABLE, the default value is YES. You cannot change this value by specifying an OPTIONS block, even when it is specified at the SELECT level.

LENGTH

Specifies whether to use the real or maximum length for variable-length fields. This keyword applies only for variable-length fields and for the USER and DSNTIAUL output formats. It is ignored when other output formats are specified.

REAL The length of the field does not change (value of the two length bytes).

MAX The output field is padded to its maximum length by adding binary zeros.

When you specify FORMAT USER, the default value for the LENGTH keyword depends on the setting of the LENGTHBYTE keyword. You can change the default value for the USER format by specifying an OPTIONS block at the GLOBAL or SELECT level or by changing the LENGTH keyword for a specific column in the FORMAT USER definition.

When you specify FORMAT DSNTIAUL, the default value is MAX. You can change the default value for the DSNTIAUL format by specifying an OPTIONS block at the SELECT level.

When you specify FORMAT VARIABLE or FORMAT DELIMITED, the default value is REAL. You cannot change this value by specifying an OPTIONS block, even when it is specified at the SELECT level.

When you specify FORMAT EXTERNAL, the default value is MAX. You cannot change this value by specifying an OPTIONS block, even when it is specified at the SELECT level.

NULLID

Specifies whether a null indicator byte is to be added at the beginning of an output field. You can also specify NULLID in the SELECT statement in the FORMAT USER syntax.

YES The null indicator is created. If the column is null, the indicator is X'FF'. Otherwise, the indicator is X'00'. The DB2 LOAD command uses the indicator to load null values into a table. You can change the values of the null indicator by specifying the NULL keyword.

NO The null indicator is not created.

The default values for formats are shown in "OPTIONS block syntax and description" on page 134.

NULLPOS

Specifies the position of the NULL indicator. You can also specify NULLPOS in the SELECT statement in the FORMAT USER syntax. This parameter does not apply to a LOB column unloaded in a spanned format (SPANNED YES option). The NULL indicator is written before the output field (containing the data) whatever the value of NULLPOS.

BEFORE

The null indicator is placed before the data field.

AFTER

The null indicator is placed after the data field.

The default value of the NULLPOS option that is used depends on the output format that is used, as shown in "OPTIONS block syntax and description" on page 134.

LOADINDDN

Specifies whether the INDDN ddname card is inserted into the DB2 LOAD command.

YES Generates the INDDN ddname card into the DB2 LOAD command. The ddname points to the data set that contains the unloaded data.

NO Does not generate the INDDN card into the DB2 LOAD command.

The default value is either NO or the value that is specified in the VUU030/ULOPTNS parameter, with parameter LOADINDDN.

FLOAT

Specifies the output format of the numeric floating point data. The FLOAT keyword applies only to the binary output format.

S390 Indicates that the binary floating point data is written to the output records in the S/390[®] internal format (hexadecimal floating point or HFP).

IEEE Indicates that the binary floating point data is written to the output records in the IEEE format (binary floating point or BFP).

The default value is S390.

Restriction: The same output format is used for all FLOAT columns in one SELECT statement. You cannot specify the FLOAT keyword at the column level in the USER format.

UNLROWSET

Specifies the number of rows that a single SQL FETCH statement retrieves.

Specifying the number of rows to retrieve improves the performance of SELECT statements that are processed by DB2 when you specify DB2 FORCE or DB2 YES with SELECT statements that are not supported by DB2 HPU. Valid values are 1 - 32767. When you specify 1, a single-row FETCH statement is used instead of a multiple-row FETCH statement. The multiple-row FETCH statement is available only in DB2 Version 8 NFM or later. The UNLROWSET keyword is ignored for earlier DB2 versions.

The default value is the value of the VUU035/ULROWSET parameter.

NULLPAD

Specifies whether the padding character is used to fill the output field when a null value is unloaded.

NO When a null value is unloaded, the output field is filled with binary zeros (x'00').

YES When a null value is unloaded, the output field is filled with the padding character.

The default values for formats are shown in “OPTIONS block syntax and description” on page 134.

When you specify FORMAT DELIMITED, the value of the NULLPAD keyword is always YES, and you cannot change it.

AUTOTAG *size*

Allows you to add a *tag*, which is an external numeric value, to each record that is unloaded. The output data for each SELECT statement has its own tag. If you specify AUTOTAG at the GLOBAL level or at the UNLOAD level, the AUTOTAG keyword applies to all SELECT statements if they are written in a common file.

If you specify AUTOTAG at the SELECT level, DB2 HPU requires all SELECT statements that write in a common output file to use the same number of digits. File records with an autotag cannot be mixed with file records without an autotag.

The generated tag starts at 1 and is automatically incremented for each SELECT statement of the same SYSIN for which AUTOTAG is either explicitly coded or generated by a LISTDEF.

The AUTOTAG keyword applies only to logical unloads, which use SELECT statements to unload DB2 data. It does not apply to physical unloads.

size Specifies the number of digits in the tag. Valid values are 0 - 8. If you specify 0, an AUTOTAG is not generated for the corresponding SELECT statements.

Important: If the value of the tag that is computed for a SELECT statement exceeds the number of digits that you specified for the AUTOTAG keyword, DB2 HPU stops.

If you specify AUTOTAG and LOADDDN is requested, the LOADDDN file that is generated automatically includes a WHEN clause for each SELECT statement to reload into the corresponding tables. In this case, the unloaded tables correspond with the value of the autotag.

The default value is the value of the VUU046/ULAUTAG parameter.

IFERROR

Defines the value that is assigned to an output field if a conversion error occurs while the output field is formatted or if an error occurs while an expression that is natively processed by DB2 HPU is evaluated. The IFERROR keyword applies only to conversion or formatting that was made during the unload process. The IFERROR keyword does not apply to conversion errors when constants are specified in the SELECT statement and these constants are converted into a different data type by using an INTO clause, a REFORMAT clause, or by specifying FORMAT USER. These constants are converted during the initialization process. If an error occurs during this process, a syntax error is issued, and processing stops.

The value that you specify applies to all output fields for the corresponding SELECT statements.

If an expression error for a SELECT statement that is processed by DB2 occurs, DB2 issues an SQL error message, and the default value does not apply.

DEFAULT

The default value is assigned. This value depends on the data type.

NULL_OR_DEFAULT

If the output field allows a null value, the NULL value is assigned to the field. Otherwise, the default value is used.

TRIM Allows you to suppress the trailing blank characters when data is unloaded into VARCHAR, VARGRAPHIC, output fields, or CLOB, and DBCLOB output fields expect if SPANNED YES is used. The length of the output field is adjusted to match the effective number of characters that are written.

If you specify FORMAT DELIMITED, the TRIM keyword applies to output fields with CHAR and GRAPHIC data types.

If you specify the TRIM keyword at the GLOBAL level, it applies only to the USER format unless YES is specified for the VUU057/OPALLFMT PARMLIB parameter. If you specified YES for the VUU057/OPALLFMT PARMLIB parameter or if you specify the TRIM keyword at the SELECT level, the TRIM keyword applies to any output format.

The TRIM keyword does not apply when numeric data is converted to numeric external.

The default value is the value of the VUU059/ULTRIM PARMLIB parameter.

PADDING

Specifies the padding character that DB2 HPU uses at the column level or the record level.

Column

The padding character is used to pad the null values when you specify NULLPAD YES to pad the output field when the specified value is shorter than the field or to pad numeric data that is converted to character data.

Record

The padding character is used to pad the end of the output record up to the length of the physical record when the format of the DCB (RECFM) is fixed and when the output record is shorter than the LRECL.

The following syntaxes are supported:

PADDING (SBCS padding [, DBCS padding [, record padding]])

Specifies the padding characters for SBCS and DBCS characters when a character string is converted to a longer character string and when the generated record is padded to its specified length.

SBCS padding

The default value is ' ' (an EBCDIC SBCS blank character). You can specify the value in the following ways:

'c' Specifies an EBCDIC SBCS character. This character can be converted to match the output CCSID.

X'hh' Specifies an SBCS character in hexadecimal format. This character is used as-is, regardless of the output CCSID.

DBCS padding

The default value is the DBCS space character in the output CCSID. To obtain this DBCS space character, the UTF16 character UX'3000' is converted into the output CCSID. You can specify the value in the following ways:

X'hhhh' Specifies a DBCS character in a 2-byte hexadecimal format. This character is used as-is, regardless of the output CCSID.

G'SI_graphic_char_SO' Specifies a graphic character as a graphic constant that is expressed in EBCDIC. Shift-out/shift-in bytes are included. This character can be converted to match the output CCSID.

GX'hhhh' Specifies a graphic character as a hexadecimal EBCDIC DBCS constant. This character can be converted to match the output CCSID.

UX'hhhh' Specifies a graphic character as a hexadecimal UTF16 Unicode constant. This character can be converted to match the output CCSID.

Record padding

Depending on the output format that is used to unload the data, the default value can be either of the following values:

- X'00' for the DSNTIAUL format and the EXTERNAL format
- The SBCS blank character that was converted in the output CCSID for all other formats

You can specify the value in the following ways:

'c' Specifies an EBCDIC SBCS character. This character can be converted to match the output CCSID.

X'hh' Specifies an SBCS character in hexadecimal format. This character is used as-is, regardless of the output CCSID.

If you do not specify the PADDING keyword, the following values are default values for padding at the character level and the record level:

SBCS characters

- X'00' for variable-length columns in the DSNTIAUL format and the EXTERNAL format
- The SBCS blank character that was converted to match the output CCSID for all other formats and, if padding at the column level is not specified, for columns that use an INTO clause or a REFORMAT keyword

DBCS characters

UTF16 value UX'3000' that was converted to match the output CCSID

Record padding character

- X'00' for the DSNTIAUL format and the EXTERNAL format
- The SBCS blank character that was converted to match the output CCSID for all other formats

Important: When you specify FORMAT DSNTIAUL or FORMAT EXTERNAL, PADDING () does not give the same result as omitting the PADDING keyword because the SBCS padding is x'00' when the PADDING keyword is omitted, and it is the SBCS blank character when the PADDING keyword is specified.

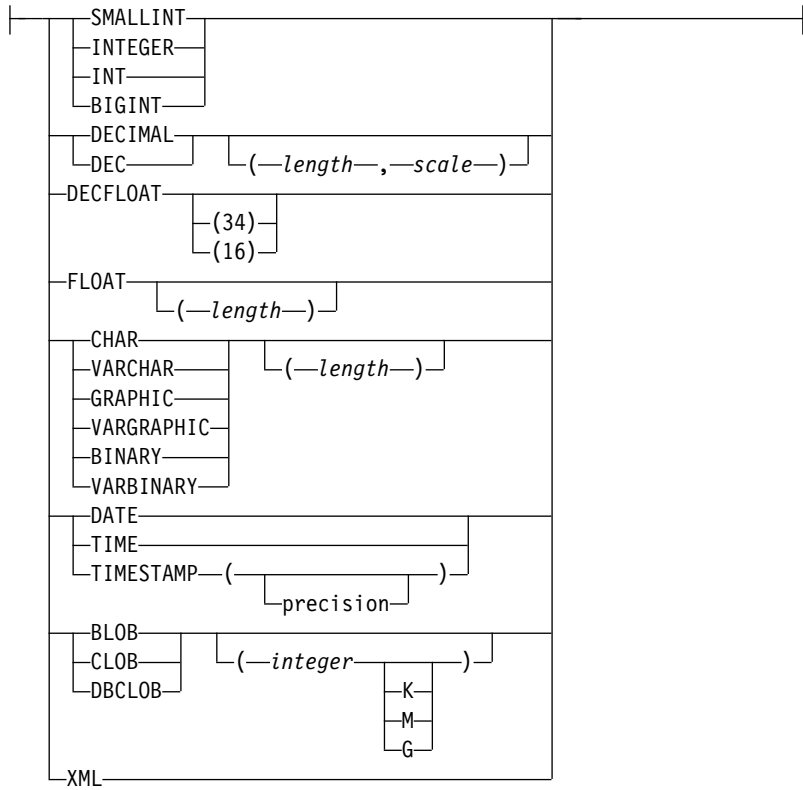
REFORMAT

Defines the default reformatting for all columns that correspond to a specific DB2 definition.

internal-data-type

Specifies a DB2 data type as it is specified in the CREATE TABLE statement. For columns that allow a length or length and scale specification, such as CHAR or DECIMAL columns, you can specify the data type with or without the length. Specifying the data type without the length defines a default conversion for all columns with this data type. If you specify a data type plus a length, such as CHAR(8), the default conversion applies only to columns that are defined with this precise data type. The following diagram shows the syntax of the internal data type:

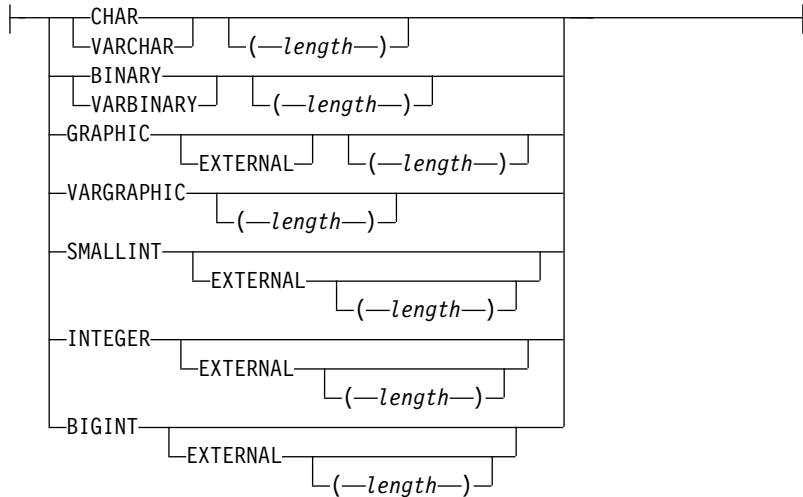
internal-data-type:

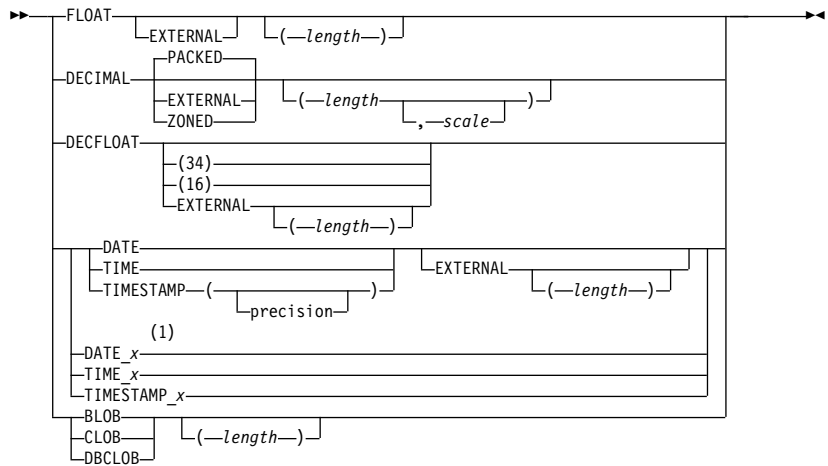


output-data-type

The following syntax diagram shows the output data type:

output-data-type:





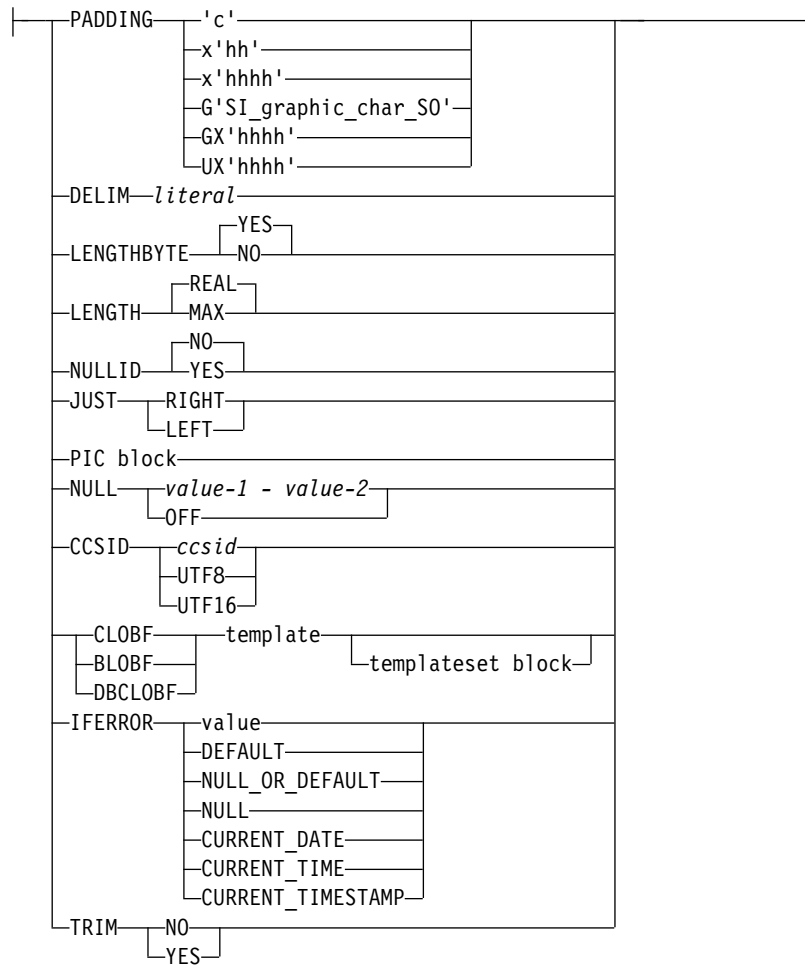
Notes:

- 1 The DATE, TIME, and TIMESTAMP data types without the EXTERNAL keyword correspond to the internal DATE/TIME/TIMESTAMP format, and DATE_x, TIME_x, and TIMESTAMP_x correspond to the formats that are defined in the OPTIONS description.

column-format-option

Use the column format option block in the REFORMAT clause of the OPTIONS block, in the FORMAT USER statement, or in the INTO clause of the SELECT statement. The following diagram shows the syntax of the column format option:

column-format-option:



For example, you can use the REFORMAT keyword to specify that you want all CLOB columns that are processed are changed into CLOBF columns by using a specific TEMPLATE statement without having to define it in an INTO clause or in a FORMAT USER statement. You can also use the REFORMAT keyword to convert all DECIMAL(10,0) columns in INTEGER format.

The REFORMAT keyword is useful when you use LISTDEF to make format conversions because the INTO clause is not supported and using the USER format is not possible.

The formatting options that are specified in the REFORMAT keyword are overridden by the options that are defined in the INTO clause of the fast select block by the LIKE keyword that is used in the DSNTIAUL format or the VARIABLE format and by the conversion option that is defined in the USER format.

PADDING *padding-character*

Specifies the padding character, where *padding-character* is the padding character to be used when padding is required for a column.

Padding is used to convert characters to a longer string. Padding is typically applied to the end of character strings. If you specify

JUST RIGHT, padding is added at the beginning of the string. For SBCS data, you can specify the padding character in the following ways:

- 'c' Specifies an EBCDIC SBCS character. This character can be converted to match the output CCSID.
- X'hh' Specifies an SBCS character in hexadecimal format. This character is used as-is, regardless of the output CCSID.

For DBCS data, you can specify the padding character in the following ways:

- X'hhhh' Specifies a DBCS character in a 2-byte hexadecimal format. This character is used as-is, regardless of the output CCSID.
- G'SI_graphic_char_SO' Specifies a graphic character as a graphic constant that is expressed in EBCDIC. Shift-out/shift-in bytes are included. This character can be converted to match the output CCSID.
- GX'hhhh' Specifies a graphic character as a hexadecimal EBCDIC DBCS constant. This character can be converted to match the output CCSID.
- UX'hhhh' Specifies a graphic character as a hexadecimal UTF16 Unicode constant. This character can be converted to match the output CCSID.

You must specify a single-byte character ('c' or X'hh') for SBCS data and a double-byte character (G'SI_graphic_char_SO' or GX'hhhh' or UX'hhhh') for DBCS data. Any other combination is rejected. The following examples show how you can specify padding:

- PADDING '*'
- PADDING X'00' (binary zero)

For SBCS data, the default padding character is an SBCS blank character that is converted in the output CCSID, if needed. For DBCS data, the default padding character is the UTF16 character UX'3000' that is converted in the output CCSID, if needed.

DELIM *literal*

When this keyword specified in a USER block, it indicates the delimiter to be used in external DATE or TIME fields. The literal must be a single character and must be one byte long, regardless of the literal CCSID.

For DATE fields, a hyphen (-) is the default value.

For TIME fields, a period (.) is the default value.

For the TIMESTAMP column, both delimiters are used.

LENGTHBYTE

Specifies whether to write the two length bytes for variable-length columns to the output data set.

YES The two length bytes are written.

NO The two length bytes are not written.

The default value is YES.

LENGTH

Specifies whether the real or maximum length is to be used for variable-length fields.

REAL The length of the field does not change (value of the two length bytes).

MAX The output field is padded to its maximum length with binary zeros.

The LENGTH keyword is useful only for variable-length fields.

The default value is REAL.

NULLID

Specifies whether a null indicator byte is added to the beginning of an output field. You can specify NULLID in the OPTIONS block also.

YES The null indicator is created. If the column is null, this indicator is set to the X'FF' value. Otherwise, the indicator is set to X'00'. The indicator can be used by the DB2 LOAD command to load null values into a table. You can change values of the null indicator by using the NULL keyword.

NO The null indicator is not created.

The default value is YES.

JUST Specifies whether to align the output character string. The JUST keyword specifies right or left justification for extended numeric values or for character strings when strings are converted to a greater length.

RIGHT

Justify the output character string to the right.

LEFT Justify the output character string to the left.

The default value depends on the type of string. For conversion between character strings, the default value is LEFT. For numeric conversions in strings, the default value is RIGHT.

PIC block

For a description of the PIC block, see the PIC keyword.

NULL Indicates whether the null indicator is generated in the output data set. You can specify NULL in the OPTIONS block also.

value-1

Indicates the value of the null indicator when the column value is NULL. It can be specified in character ('C') or hexadecimal (X'hh') format.

value-2

Indicates the value of the null indicator when the column value is NOT NULL. It can be specified in character ('c') or hexadecimal (X'hh') format.

OFF No null indicator is generated.

The default is the value that is specified for the VUU014/ULNULL PARMLIB parameter.

CCSID

Specifies that a CCSID conversion will be done on the output field. The CCSID conversion between the CCSID of the unloaded data and the CCSID that was specified in this parameter must be defined. When a LOB file reference is specified, the specific CCSID conversion is applied on the LOB data. The LOBFILE DSNNAME that is generated in the output record is kept in the default CCSID of the output file.

ccsid Specify a valid CCSID value for the CCSID of the output field.

UTF8 Unicode Transformation Format, 8-bit encoding form is used. UTF8 is equivalent to 1208.

UTF16 Unicode Transformation Format, 16-bit encoding form is used. UTF16 is equivalent to 1200.

BLOBF

Specifies that the output field will contain the name of the file in which the BLOB is to be unloaded without CCSID conversion.

CLOBF

Specifies that the output field will contain the name of the file in which the CLOB will be unloaded with any required CCSID conversion.

DBCLOBF

Specifies that the output field will contain the name of the file in which the DBLOB will be unloaded with any required CCSID conversion.

template-statement-name

Specifies the name of a TEMPLATE statement that is used to allocate the file that will contain the LOB data. The DSN of the TEMPLATE statement that is used must contain either the &UNIQ variable that is explicitly coded or automatically generated for a PDS or PDS/E, or a user-defined variable that is associated with a TEMPLATESET to the :RECNUM predefined variable.

BLOBF, CLOBF, and DBCLOBF output types can be used only when converting BLOB, CLOB and DBCLOB columns. The output type that is defined with the TYPE keyword must be either CHAR or VARCHAR, and the corresponding column must be large enough to contain the generated data set name.

To use LOB file reference with a table that comes from a LISTDEF, use BLOBF, CLOBF, or DBCLOBF in the REFORMAT keyword of the OPTIONS block. This option specifies that all LOB data will be unloaded by using LOB file reference. The associated TEMPLATE statement must distinguish the generated files to prevent duplicates. When you unload data from a LISTDEF with tables that

contain potentially more than one LOB column, the TEMPLATE statement that is used in the REFORMAT keyword can contain a user-defined variable that is associated with the :COLNUM or :COLNAME predefined variables. When you specify the &TS or &SN variables in a TEMPLATE statement that is used with BLOBF, CLOBF, or DBCLOBF, DB2 HPU substitutes the name of the table space that stores the LOB column value, not the base table space name. By substituting the name of the table space, DB2 HPU can generate unique data set names for each LOB column.

IFERROR

Defines the value that must be assigned to the output field if a conversion error occurs when the output field is formatted or if an error occurs when an expression is evaluated by DB2 HPU. IFERROR applies only to conversion or formatting that was made during the unload process. IFERROR does not apply to conversion errors when constants are specified in the SELECT statement and these constants are converted into a different data type by using an INTO clause, a REFORMAT keyword, or a FORMAT USER specification. These constants are converted during the initialization process. If an error occurs during this process, a syntax error is issued, and processing stops.

If an expression error for a SELECT statement that was processed by DB2 occurs, DB2 issues an SQL error message, and a default value does not apply.

value Specifies a value to assign to the output field in a conversion error or an error that occurs when an expression is evaluated. The following constant values are valid:

Character or hexadecimal constants

Character constants or hexadecimal constants are accepted for any type of output field. In CCSID conversion, character constants are converted from EBCDIC into the output CCSID, but hexadecimal constants are not converted into the output CCSID.

When you use the following output data types, DB2 HPU uses CCSID conversion for character constants:

- CHAR
- VARCHAR
- CLOB
- GRAPHIC
- GRAPHIC EXTERNAL
- VARGRAPHIC
- DBCLOB
- SMALLINT EXTERNAL
- INTEGER EXTERNAL
- BIGINT EXTERNAL
- FLOAT EXTERNAL
- DECIMAL EXTERNAL
- DECFLOAT EXTERNAL

- DATE_EXTERNAL
- DATE_A to DATE_P
- DATE_DB2
- TIME_EXTERNAL
- TIME_A to TIME_E
- TIME_DB2
- TIMESTAMP_EXTERNAL
- TIMESTAMP_A to TIMESTAMP_G

When you use the following output data types, DB2 HPU does not use CCSID conversion for character constants:

- BINARY
- VARBINARY
- BLOB
- SMALLINT
- INTEGER
- BIGINT
- FLOAT
- DECIMAL PACKED
- DECIMAL ZONED
- DECFLOAT
- DATE
- DATE_Q - DATE_R
- DATE_0 - DATE_7
- TIME
- TIME_0
- TIMESTAMP
- TIMESTAMP_0.

Numeric constants

Use numeric constants only if the output field corresponds to numeric data in internal format. Otherwise, use a character constant.

The following table shows the conversions that are accepted:

Table 25. Accepted conversion types

Output data type	Type of numeric constant accepted
SMALLINT	Integer constant
INTEGER	Integer constant
BIGINT	Integer or BIGINT constant
DECIMAL	Integer constant that fits in the DECIMAL output field or a decimal constant with scale and precision that is smaller or equal to the scale and precision of the output field
FLOAT	Integer, BIGINT, decimal, or float constants
DECFLOAT	Any numeric type

DEFAULT

The default value is assigned to the field. The default value depends on the data type.

NULL_OR_DEFAULT

If the output field allows a null value, the NULL value is assigned to the field. Otherwise, the default value is assigned.

NULL The NULL value is assigned to the field. The output field must include a null indicator.

CURRENT_DATE, CURRENT_TIME, CURRENT_TIMESTAMP

The assigned date, time, or timestamp is identical for each row, and its value is obtained at the beginning of the process. In internal or external format, CURRENT_DATE is allowed only for the DATE field; CURRENT_TIME is allowed only for the TIME field; and CURRENT_TIMESTAMP is allowed only for the TIMESTAMP field.

TRIM Allows you to suppress the trailing blank characters when you unload data into VARCHAR, VARGRAPHIC, CLOB, and DBCLOB output fields. If you specify FORMAT DELIMITED, the TRIM keyword applies to output fields with CHAR and GRAPHIC data types. The length of the output field is adjusted to match the effective number of characters that are written.

The TRIM option does not apply when numeric data is converted to numeric external.

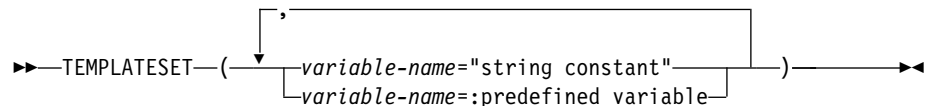
NO Trailing blank characters are not suppressed.

YES Trailing blank characters are suppressed.

The default value is the value of the VUU059/ULTRIM PARMLIB parameter.

TEMPLATESET

Use the TEMPLATESET block to assign values to any user-defined variable that is used in a TEMPLATE definition. Assign values by using string constants or predefined variables that are set by DB2 HPU at run time. A colon (:) must precede predefined variable names. The following diagram shows the syntax of the TEMPLATESET block:



A predefined variable can be one of the following variables:

:COLNUM

Specifies the position of the column in the SELECT statement. This variable is a 5-digit numeric value and is generally used when LOB columns are unloaded by using a LOB file reference (CLOBF, BLOBF, or DBLOBF data types).

:SELNUM

Specifies the sequential number of the SELECT statement in the UNLOAD command. This variable is a 5-digit numeric value.

:UNLNUM

Specifies the sequential number of the UNLOAD command. This variable is a 5-digit numeric value.

:RECNUM

The record number is a 15-digit numeric value and is internally processed. By default, the variable that is associated with :RECNUM is replaced by the last seven digits of the record number. A substring can be applied to this variable to obtain another part of the record number, or to retrieve fewer digits. This variable is generally used when LOB columns are unloaded by using a LOB file reference (CLOBF, BLOBF, or DBLOBF data types).

:COLNAME

This predefined variable contains the first characters of the column name. When the associated variable is replaced by its value, it is truncated to make it a valid qualifier in the generated DSNNAME. Valid values are 1 - 8 characters. This variable is generally used when LOB columns are unloaded by using a LOB file reference (CLOBF, BLOBF, or DBLOBF data types).

When these variables cannot be evaluated, :COLNUM and :SELNUM are set to 00000; :RECNUM is set to 0000000; and :COLNAME is set to COL n , where n is a sequential number. This sequential number ensures unicity, but it does not have a specific meaning regarding the actual database or the table space that it represents.

:SELNUM, does not apply when the TEMPLATE statement is used with UNLDDN.

:COLNUM, :COLNAME, and :RECNUM, do not apply when the TEMPLATE statement is used outside of a LOB file reference.

You can code the TEMPLATESET block in the OPTIONS or GLOBAL OPTIONS block. When the TEMPLATESET block is used with LOB columns that use a LOB file reference, the TEMPLATESET block can be used in the column format option block. The COLUMN FORMAT OPTIONS block can be used in the REFORMAT keyword, in the INTO clause, or in the FORMAT USER definition.

All TEMPLATESET definitions that are coded in high-level syntax are available at a lower level. For example, if TEMPLATESET is coded to define a user variable in the GLOBAL OPTIONS block, this variable will be available to build data set names in the templates that are used in all UNLOAD commands and for all SELECT statements in these UNLOAD commands. However, if TEMPLATESET is defined only in an OPTIONS statement that is coded in a SELECT statement, the variable definition will be available only for the TEMPLATE statements that are used in this specific SELECT statement. The definition of a user variable with a TEMPLATESET that is made at a lower level overrides the definition of the same user-defined variable that was made at a higher level but only at this lower level. For example, if a user variable is set to :SELNUM in the GLOBAL OPTIONS block and set to ABC in the OPTIONS statement that is coded in one of the SELECT statements, the replaced value will be :SELNUM for all other SELECT statements, and ABC for the SELECT statement that overrides the variable definition.

HIDDEN

Specifies whether hidden columns are unloaded when a SELECT *

statement is used. You can also specify this keyword as the third parameter after the UTILID in the PARM field in the EXEC statement. When the HIDDEN parameter is specified and this keyword is specified in SYSIN, the value that is specified in SYSIN is used.

- NO** Specifies that hidden columns are not unloaded when a SELECT * statement is used.
- YES** Specifies that hidden columns are unloaded when a SELECT * statement is used. For SELECT statements that are processed by DB2, the HIDDEN keyword is ignored.

If the HIDDEN keyword is not specified in the SYSIN, the value for HIDDEN that is specified on the EXEC card is the default value. If the HIDDEN keyword is not specified on the EXEC card, the default value is the value of the VUU042/ULHIDDEN PARMLIB parameter.

SPANNED

For logical unload statement (using a SELECT clause) which result table has LOB columns and one of the CLOB, BLOB or DBCLOB format is applied to these columns. This option specifies whether the output data must be unloaded into a VBS data set in

spanned record format compliant with the DB2 LOAD utility used with the FORMAT SPANNED YES option.

- NO** The records of the output data set do not have the spanned format.
- YES** The records of the output data set have the spanned format. The content of the LOB data is unloaded along with the data from the other columns.

Note:

- SPANNED YES only applies to LOB data being unloaded as LOB but not if a conversion to any other type is requested.
- SPANNED YES can only be used when the FORMAT option specifies one of the following:
 - VARIABLE
 - USER provided the related LOAD statement can be generated
- SPANNED YES cannot be used with DSNTIAUL, DELIMITED, INTERNAL and EXTERNAL format.
- when SPANNED YES is specified, the parameter TRIM YES does not apply for LOB columns.

ENFORCE_COLUMN_ORDER

If SPANNED YES option is requested, the LOB or XML data will be placed at the end of the result table (defined by the column list of the SELECT clause and by the optional INTO clause). This condition might not be fulfilled because of the sequence of the explicit expressions that define the result table or because of the underlying table definition when a SELECT* is specified. DB2 HPU can rearrange the columns of the result table to ensure the LOB columns are gathered at the end of the record. The ENFORCE_COLUMN_ORDER specifies if the reordering feature is disabled, i.e. if the native column sequence of the result table is kept.

- YES** The order of the column in the select statement is kept. If the LOB columns are not the last (rightmost) columns of the result table, the processing stops and an error message is issued.

TBNAME

This variable is substituted with the following value according to the precedence order of the list below:

- the table name specified via the INTO clause ;
- the name of the table to be unloaded ;
- "TBnnn" where nnn is a unique number of non of the above can be determined.

COLNAME

This variable is substituted with the following value according to the precedence order of the list below:

- "UNDEFINED" if not applicable (i.e. used to specify the RECORD xmltag) ;
- the column name specified via the INTO clause ;
- the name of the column to be unloaded ;
- "COLnnn" where nnn is a unique number of none of the above can be determined.

When these variables cannot be evaluated, :COLNUM and :SELNUM are set to 00000; :RECNUM is set to 0000000; and :COLNAME is set to COL*n*, where *n* is a sequential number. This sequential number ensures unicity, but it does not have a specific meaning regarding the actual database or the table space that it represents.

:SELNUM, does not apply when the TEMPLATE statement is used with UNLDDN.

:COLNUM, :COLNAME, and :RECNUM, do not apply when the TEMPLATE statement is used outside of a LOB file reference. You can code the TEMPLATESET block in the OPTIONS or GLOBAL OPTIONS block. When the TEMPLATESET block is used with LOB columns that use a LOB file reference, the TEMPLATESET block can be used in the column format option block. The COLUMN FORMAT OPTIONS block can be used in the REFORMAT keyword, in the INTO clause, or in the FORMAT USER definition.

All TEMPLATESET definitions that are coded in high-level syntax are available at a lower level. For example, if TEMPLATESET is coded to define a user variable in the GLOBAL OPTIONS block, this variable will be available to build data set names in the templates that are used in all UNLOAD commands and for all SELECT statements in these UNLOAD commands. However, if TEMPLATESET is defined only in an OPTIONS statement that is coded in a SELECT statement, the variable definition will be available only for the TEMPLATE statements that are used in this specific SELECT statement. The definition of a user variable with a TEMPLATESET that is made at a lower level overrides the definition of the same user-defined variable that was made at a higher level but only at this lower level. For example, if a user variable is set to :SELNUM in the GLOBAL OPTIONS block and set to ABC in the OPTIONS statement that is coded in one of the SELECT statements, the replaced value will be :SELNUM for all other SELECT statements, and ABC for the SELECT statement that overrides the variable definition.

Table 26. Variables usable in xml-templates

Variable	Alias	Description	Scope	Default
UNLNUM	UNLN	Number of the UNLOAD command as affected by DB2 HPU	RECORD & COLUMN	No default
DBNAME	DB	Name of the data base to which the unloaded table belongs	RECORD & COLUMN	DBnn(*)
TSNAME	TS	Name of the table space to which the unloaded table belongs	RECORD & COLUMN	TSnn(*)
SELNUM	SELN	Number of the SELECT statement as affected by DB2 HPU	RECORD & COLUMN	No default
TBCREATOR	TBC	Name of the creator of the unloaded table	RECORD & COLUMN	CRnn(*)
TBNAME	TBN	Name of the unloaded table	RECORD & COLUMN	TBnn(*)
COLNAME	CNAM	Name of the unload column	COLUMN	COLnnn where nnn is the rank of the column in the SELECT statement
COLNUM	CNUM	Rank of the column in the SELECT statement	COLUMN	-1
COLTYPE	CTYP	DB2 data type of the unloaded column	COLUMN	UNDEF
COLLENGTH	CLEN	Length of the unloaded column	COLUMN	-1
COLPREC	CPRE	Precision of the column. Applies only to a decimal column	COLUMN, type decimal	-1
COLSCALE	CSCA	Scale of the column. APplies only to a decimal column	COLUMN, type decimal	-1
COLNULL	CNUL	Nullability attribute of the column (the value is either NULL, NOT NULL or UNDEF if DB2 HPU cannot determine the attribute.	COLUMN	UNDEF

OPTIMIZATION_HINT

Allows to process an SQL statement: SET CURRENT OPTIMIZATION HINT 'string constant' when processing the unload in SQL mode. This option is only available if DB2 parameter is set to FORCE. For example, in DB2 HPU syntax, OPTIMIZATION_HINT 'NOHYB' generates the SQL statement: SET CURRENT OPTIMIZATION HINT = 'NOHYB'

You can clear the CURRENT OPTIMIZATION HINT special register by specifying an empty string. For example, in DB2 HPU syntax: OPTIMIZATION_HINT generates the SQL statement: SET CURRENT OPTIMIZATION HINT = ' '

QUERY_ACCELERATION value

Allows to have the following SQL statement run when a logical unload is processed in SQL mode:

```
SET CURRENT SQL ACCELERATION = 'value'
```

The value can be one of the following values: NONE, ENABLE, ENABLE WITH FAILBACK, ELIGIBLE and ALL. It refers to the relevant DB2 publication for a description of the effect of the QUERY_ACCELERATION special register.

Important: This option is only available if DB2 parameter is set to FORCE

Note: If QUERY_ACCELERATION parameter is specified (except for NONE value) and if the parameter UNLROWSET or its default VUU0035/ULROWSET is set to 1, the multi row fetch mode is forced to 100 rows.

Related concepts:

“Output encoding scheme” on page 65

For the DSN TIAUL, DELIMITED, VARIABLE, USER, and EXTERNAL output formats, the translations from EBCDIC to ASCII and from ASCII to EBCDIC are supported only for single-byte character set (SBCS) character strings. Data is translated by using the translation tables in the SYSIBM.SYSSTRINGS table.

Related reference:

“FORMAT block syntax and description” on page 190

Use the FORMAT block to specify the format of the data that is unloaded. The FORMAT block is a part of the SELECT block.

“TIMESTAMP format types” on page 461

Use the TIMESTAMP format type to specify the output data format.

“TIME format types” on page 461

Use the TIME format type to specify the output data format.

“DATE format types” on page 459

Use the DATE format type to specify the output data format.

Example: Unloading in XML format from table columns only with the additional AUTOTAG field

In this example, you can unload in XML format from table column with the additional AUTOTAG field.

Unload with records tagged with 'HEADER' and columns tagged with their name. An extra field is requested via the AUTOTAG options. As no column name is available for this extra field, the COL000 tag is used for this field. The NULLL values are replaced by 'NULL'.

UNLOAD statement:

```
UNLOAD TABLESPACE
```

```
OPTIONS AUTOTAG 2
```

```
SELECT DEPTNO, MGRNO, ADMRDEPT FROM DSN8910.DEPT
FORMAT XML DELIM ''' NULLVAL 'NULL'
OUTDDN (FOUT)
OPTIONS
XMLSET ( COLUMN = < &COLNAME. > , RECORD = < HEADER > )
```

Output:

```
<HEADER><COL000>01</COL000><DEPTNO>"D01"</DEPTNO><MGRNO>NULL</MGRNO><ADMRDEPT>"A00"</ADMRDEPT></HE
<HEADER><COL000>01</COL000><DEPTNO>"D11"</DEPTNO><MGRNO>"000060"</MGRNO><ADMRDEPT>"D01"</ADMRDEPT>
```

Example: Unloading in XML format with an SQL statement involving SQL expressions from an identified table

In this example, you unload data in XML format with an SQL statement involving SQL expressions from an identified table.

Unload with records tagged with 'HEADER' and columns tagged with their name. An extra field is requested via the AUTOTAG options and an SQL expression is involved by the SELECT. As no column name is available for both the AUTOTAG field and the SQL expression, the COL000 and COL001 tags are used for these fields. The NULLL values are replaced by 'NULL'.

UNLOAD statement:

```
UNLOAD TABLESPACE
```

```
OPTIONS AUTOTAG 2
```

```
SELECT DEPTNO, CONCAT (MGRNO,MGRNO) FROM DSN8910.DEPT
FORMAT XML DELIM '"' NULLVAL 'NULL'
OUTDDN (FOUT)
OPTIONS
XMLSET ( COLUMN = < &COLNAME. > , RECORD = < &TBNAME > )
```

Output:

```
<DEPT><COL000>01</COL000><DEPTNO>"A00"</DEPTNO><COL002>"000010000010"</COL002></DEPT>
<DEPT><COL000>01</COL000><DEPTNO>"B01"</DEPTNO><COL002>"000020000020"</COL002></DEPT>
<DEPT><COL000>01</COL000><DEPTNO>"C01"</DEPTNO><COL002>"000030000030"</COL002></DEPT>
<DEPT><COL000>01</COL000><DEPTNO>"D01"</DEPTNO><COL002>NULL</COL002></DEPT>
```

Example: Unloading in XML with an SQL statement involving SQL expressions and an INTO clause

In this example, you unload data in XML format with an SQL statement involving SQL expressions and an INTO clause

Unload with records tagged with 'HEADER' and columns tagged with their name. An extra field is requested via the AUTOTAG options and an SQL expression is involved by the SELECT. An INTO clause requests the field name to be changed. As no column name is available for both the AUTOTAG, the COL000 is used for this field. The NULLL values are not materialized.

UNLOAD statement:

```
UNLOAD TABLESPACE
```

```
OPTIONS AUTOTAG 2
```

```
SELECT DEPTNO, CONCAT (MGRNO,MGRNO) INTO TABLE USER.NEWDEPT C1, C2 FROM DSN8910.DEPT
FORMAT XML DELIM '"'
OUTDDN (FOUT)
OPTIONS
XMLSET ( COLUMN = < &COLNAME. > , RECORD = < &TBNAME > )
```

Output:

```
<NEWDEPT><COL000>01</COL000><C1>"A00"</C1><C2>"000010000010"</C2></NEWDEPT>
<NEWDEPT><COL000>01</COL000><C1>"B01"</C1><C2>"000020000020"</C2></NEWDEPT>
<NEWDEPT><COL000>01</COL000><C1>"C01"</C1><C2>"000030000030"</C2></NEWDEPT>
<NEWDEPT><COL000>01</COL000><C1>"D01"</C1><C2><</C2></NEWDEPT>
```

Example: Unloading in XML format using variables

UNLOAD statement:

```
UNLOAD TABLESPACE
```

```
OPTIONS
```

```
XMLSET ( RECORD = < &TS. CREATOR = &CREATOR., TABLE = &TBNAME. > ,  
        COLUMN = <COLUMN NAME = &COLNAME. , TYPE = &COLTYPE. > ,
```

```
SELECT ACTNO, ACTKWD FROM DSN8910. ACT  
FORMAT XML
```

The output record is as follows:

```
<DSN8S71P CREATOR = "DSN8910" TABLE = "ACT"><COLUMN NAME = "ACTNO" TYPE =  
'SMALLINT">10<.COLUMN><COLUMN NAME = "ACTKWD" TYPE = "CHAR">'TEST'</COLUMN></DSN8S71P>
```

Example: Unloading with records tagged with 'HEADER' and columns tagged with their name

Unload with records tagged with 'HEADER' and columns tagged with their name. An extra field is requested via the AUTOTAG options. As no column name is available for both the AUTOTAG, the COL000 is used for this field. The NULL values are replaced by 'NULL'.

UNLOAD statement:

```
UNLOAD TABLESPACE
```

```
OPTIONS AUTOTAG 2
```

```
SELECT DEPTNO, MGRNO, ADMRDEPT FROM DSN8910.DEPT  
FORMAT XML DELIM '"' NULLVAL 'NULL'  
OUTDDN (FOUT)  
OPTIONS  
XMLSET ( COLUMN = < &COLNAME. > , RECORD = < HEADER > )
```

Output:

```
<HEADER><COL000>01</COL000><DEPTNO>"D01"</DEPTNO><MGRNO>NULL</MGRNO><ADMRDEPT>"A00"</ADPRDEPT></HEAD  
<HEADER><COL000>01</COL000><DEPTNO>"D11"</DEPTNO><MGRNO>"000060"</MGRNO><ADMRDEPT>"D01"</ADPRDEPT></HEAD
```

Unload with records tagged with 'HEADER' and columns tagged with their name. An extra field is requested via the AUTOTAG options. As no column name is available for both the AUTOTAG field and the SQL expression, the COL000 and COL001 tags are used for these fields. The NULL values are replaced by 'NULL'.

UNLOAD statement:

```
UNLOAD TABLESPACE
```

```
OPTIONS AUTOTAG 2
```

```
SELECT DEPTNO, CONCAT (MGRNO,MGRNO) FROM DSN8910.DEPT  
FORMAT XML DELIM '"' NULLVAL 'NULL'  
OUTDDN (FOUT)  
OPTIONS  
XMLSET ( COLUMN = < &COLNAME. > , RECORD = < &TBNAME. > )
```

Output:

```
<DEPT><COL000>01</COL000><DEPTNO>"A00"</DEPTNO><COL002>"000010000010"</COL002></DEPT>
<DEPT><COL000>01</COL000><DEPTNO>"B01"</DEPTNO><COL002>"000020000020"</COL002></DEPT>
<DEPT><COL000>01</COL000><DEPTNO>"C01"</DEPTNO><COL002>"000030000030"</COL002></DEPT>
<DEPT><COL000>01</COL000><DEPTNO>"D01"</DEPTNO><COL002>NULL</COL002></DEPT>
```

Unload with records tagged with 'HEADER' and columns tagged with their name. An extra field is requested via the AUTOTAG options and an SQL expression is involved by the SELECT. An INTO clause request the field name be changed. As no column name is available for the AUTOTAG, the COL000 is used for this field. The other fields are tagged according to the name given via the INTO clause. The NULL values are not materialized.

UNLOAD statement:

```
UNLOAD TABLESPACE
```

```
OPTIONS AUTOTAG 2
```

```
SELECT DEPTNO, CONCAT (MGRNO,MGRNO) INTO TABLE USER.NEWDEPT C1, C2
FROM DSN8910.DEPT
FORMAT XML DELIM ''' NULLVAL 'NULL'
OUTDDN (FOUT)
OPTIONS
XMLSET ( COLUMN = < &COLNAME. > , RECORD = < &TBNAME. > )
```

Output:

```
<NEWDEPT><COL000>01</COL000><C1>"A00"</C1><C2>"000010000010"</C2></NEWDEPT>
<NEWDEPT><COL000>01</COL000><C1>"B01"</C1><C2>"000020000020"</C2></NEWDEPT>
<NEWDEPT><COL000>01</COL000><C1>"C01"</C1><C2>"000030000030"</C2></NEWDEPT>
<NEWDEPT><COL000>01</COL000><C1>"D01"</C1><C2>NULL</C2></NEWDEPT>
```

Example: Unloading LOB data into spanned format

```
UNLOAD TABLESPACE
SELECT*
FROM me.mytable
OUTDDN (SYSREC01)*
FORMAT VARIABLE ALL
OPTIONS SPANNED YES
LOADDN LOAD
```

Examples: Setting and requesting CURRENT_QUERY_ACCELERATION

Example 1: Request the CURRENT QUERY ACCELERATION special register to be set to ENABLE for every unload performed in SQL mode by specifying the QUERY_ACCELERATION option in an option block at the TABLESPACE level.

The DB2 FORCE option ensures every unload will be performed in SQL mode.

```
UNLOAD TABLESPACE
DB2 FORCE
OPTIONS QUERY_ACCELERATION ENABLE
SELECT*
FROM SYSIBM.SYSDATABASE
OUTDDN (SYSREC)
FORMAT DSNTIAUL
```

Example 2: Request the QUERY_ACCELERATION special register to be set to ENABLE only for a specific logical unload by specifying the QUERY_ACCELERATION option in an option block at the SELECT level.

The DB2 FORCE option ensures every unload will be performed in SQL mode.

```
UNLOAD TABLESPACE
DB2 FORCE
SELECT*
FROM SYSIBM.SYSDATABASE
OUTDDN (SYSREC)
FORMAT DSNTIAUL
OPTIONS QUERY_ACCELERATION ENABLE
```

Example 3: Request the QUERY_ACCELERATION special register to be set to different values according to the related logical unload by specifying the QUERY_ACCELERATION option in an option block at the SELECT level.

The DB2 FORCE option ensures every unload will be performed in SQL mode.

```
UNLOAD TABLESPACE
DB2 FORCE
-- SELECT #1
SELECT*
FROM SYSIBM.SYSDATABASE
OUTDDN (SYSREC)
FORMAT DSNTIAUL
OPTIONS QUERY_ACCELERATION ENABLE WITH FAILBACK
-- SELECT #2
SELECT*
FROM SYSIBM.SYSTABLESPACE
OUTDDN (SYSREC)
FORMAT DSNTIAUL
OPTIONS QUERY_ACCELERATION NONE
```

Example 4: Specify a general QUERY_ACCELERATION value applicable to all logical unloads except for those having a QUERY_ACCELERATION option specified at the SELECT level.

ALL applies to the first and the third SELECT statements, ELIGIBLE for the second one and NONE for the fourth one.

```
UNLOAD TABLESPACE
OPTIONS QUERY_ACCELERATION ALL
DB2 FORCE
-- SELECT #1
SELECT*
FROM SYSIBM.SYSDATABASE
OUTDDN (SYSREC)
FORMAT DSNTIAUL
-- SELECT #2
SELECT*
FROM SYSIBM.SYSTABLESPACE
OUTDDN (SYSREC)
FORMAT DSNTIAUL
OPTIONS QUERY_ACCELERATION ELIGIBLE
-- SELECT #3
SELECT*
FROM SYSIBM.SYSUSERAUTH
OUTDDN (SYSREC)
FORMAT DSNTIAUL
-- SELECT #4
SELECT*
```

```

FROM SYSIBM.SYSDBAUTH
OUTDDN (SYSREC)
FORMAT DSNTIAUL
OPTIONS QUERY_ACCELERATION NONE

```

Example: Unloading a table space containing 3 tables and requesting the NUMRECS option to be specified in the generated LOAD statement using the actual number of unloaded rows

UNLOAD statement:

```

GLOBAL OPTIONS AUTOTAG 3
LOADOPT (,,NUMRECS &NUMRECS);
UNLOAD TABLESPACE
QUIESCE YES
--SELECT FIRST TABLE WITH 2365 ROWS
SELECT * FROM
ID6631.ID6631S1T1
OUTDDN SYSREC
LOADDDN SYSPUNCH
--SELECT SECOND TABLE WITH 15423 ROWS
SELECT * FROM
ID6631.ID6631S1T2
OUTDDN SYSREC
LOADDDN SYSPUNCH
--SELECT THIRD TABLE WITH 15 ROWS
SELECT * FROM
ID6631.ID6631S1T3
OUTDDN SYSREC
LOADDDN SYSPUNCH

```

It generates a LOAD statement where the NUMRECS option specifies a different value for each table, according to the number of rows unloaded for the related table, as follows:

```

LOAD DATA LOG NO REPLACE
EBCDIC CCSID (0037)
INTO TABLE ID6631.ID6631S1T1 NUMRECS 2365
WHEN (1:3) = '001'
(
COLINT
    POSITION (    4 )          INTEGER
    NULLIF(    8 ) = '?'
)
INTO TABLE ID6631.ID6631S1T2 NUMRECS 15423
WHEN (1:3) = '002'
(
COLCHAR
    POSITION (    4 )          CHAR (   10 )
    NULLIF (   14 ) = '?'
)
INTO TABLE ID6631.ID6631S1T3 NUMRECS 15
WHEN (1:3) = '003'
(
COLDATE
    POSITION (    4 )          DATE EXTERNAL (   10 )
    NULLIF (   14 ) = '?'
)

```

Example: Generating the access path during the data extraction using CURRENT OPTIMIZATION HINT

Assuming that string constant 'NOHYB' identifies a user-defined optimization hint in owner. PLAN_TABLE. The unload below run in SQL mode (ensured via the DB2 FORCE option) requests the CURRENT_OPTIMIZATION_HINT special register be set to the value NOHYB so that DB2 uses this optimization hint to generate the access path during the data extraction needed for this unload.

```
UNLOAD TABLESPACE
DB2 NO
SELECT *
FROM DSN81010.EMP
FORMAT DSNTIAUL
OUTDDN (OUTPUT)
```

SELECT block syntax and description

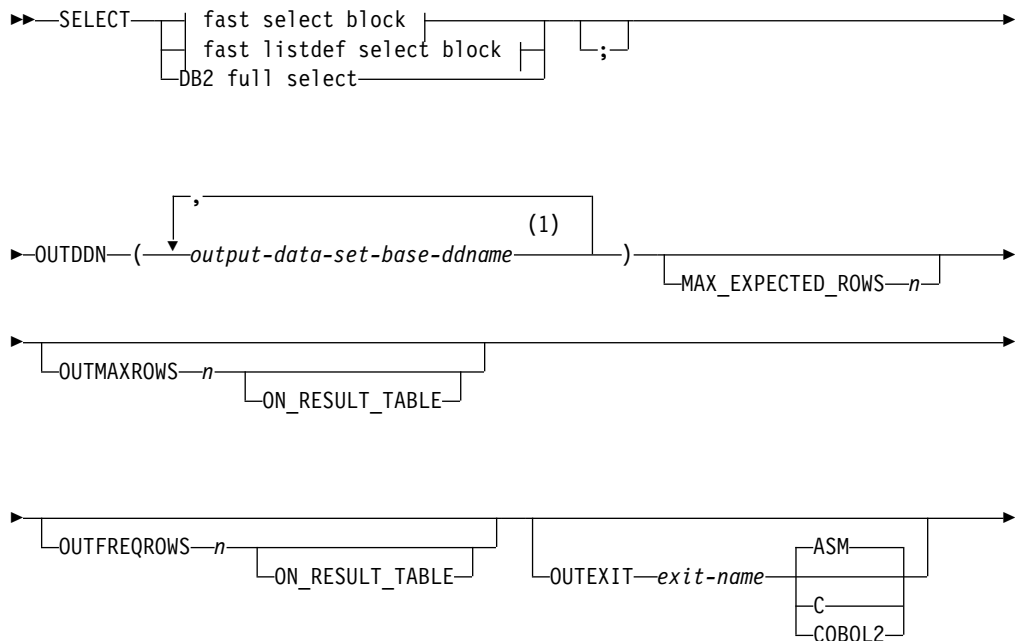
The SELECT statement specifies that a logical unload is to be done and indicates the parameters that are associated with the unload job. The SELECT block is a part of the UNLOAD block.

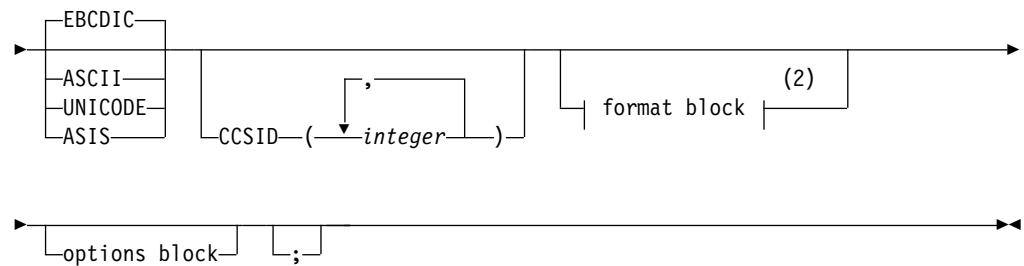
DB2 HPU supports more complex SQL SELECT statements if DB2 YES is specified in the UNLOAD block. If DB2 NO is specified, the more complex SELECT statements are passed to DB2 for processing.

If you specify more than one SELECT statement, DB2 HPU attempts to run the unload jobs in parallel.

The following diagram shows the syntax for the SELECT block:

SELECT block





Notes:

- 1 You can specify a maximum of 255 ddnames for copies of the output data set
- 2 When the format block is not specified, the default format is the value of the VUU045/ULFORMAT parameter.

SELECT

Specifies the SELECT statement as a standard SQL statement.

fast select block

See “Fast select and fast listdef select blocks syntax and description” on page 173 for keyword definitions.

fast listdef select block

See “Fast select and fast listdef select blocks syntax and description” on page 173 for keyword definitions.

DB2 full select

All valid DB2 SELECT statements that do not conform to DB2 HPU fast select block requirements are considered a DB2 full select and are processed by using DB2 unless DB2 NO was specified in the UNLOAD command.

OUTDDN *output-data-set-base-ddname*

Specifies the ddname of the sequential output data set that contains the unloaded data or the name of a TEMPLATE statement that is defined in the same SYSIN or in a TEMPLATE library. You can specify up to 255 ddnames for copies of the output data set. When you use JCL-allocated ddnames, in the JCL, include a DD statement that corresponds to each ddname that you specify.

The variable *outddn* is the base ddname of the output data set.

To process partitioned table spaces in parallel, either use a TEMPLATE with a DSNNAME that contains the &PART variable, or code in your JCL one *outddnnn* statement for each partition (*outdd01, outdd02,... outddnnn*), where *nnn* is a 1- to 7-digit sequential number that identifies a partition to be unloaded. During the unload process, data from each partition is directed to the corresponding ddname. If the corresponding ddname is allocated, it is used for the given partition. Otherwise, the base ddname is used if it is allocated.

In the following example, if MYDD, MYDD01, and MYDD0004 are allocated, MYDD contains the rows from partition 2 and 5, MYDD01 contains the rows from partition 1, and MYDD0004 contains the rows from partition 4.

```

UNLOAD TABLESPACE PART(1,2,4,5) SELECT * FROM Q.T OUTDDN(MYDD)
FORMAT DSNTIAUL

```

If a single ddname is used as output for multiple UNLOAD or SELECT statements, data from one or more of the UNLOAD or SELECT statements might be included in the single output ddname file. To ensure that data collected from each source remains together in the generated output, specify unique base ddnames for each SELECT or UNLOAD statement, . A base ddname that is used as an operand of either the UNLDDN keyword or the OUTDDN keyword is not used as a suffixed ddname, as shown in the following example:

```
UNLOAD TABLESPACE PART(1,2,4,5)
SELECT * FROM Q.TA OUTDDN(MYDD) FORMAT DSNTIAUL
SELECT * FROM Q.TB OUTDDN(MYDD1) FORMAT DSNTIAUL
```

To avoid including the data from tables Q.TA and Q.TB in the same data set, ddname MYDD1, if allocated, is used as the base ddname for only the second SELECT, and does not contain data from partition 1 of the Q.TA table. The data from partition 1 of the Q.TA table is written in ddname MYDD, which is allocated.

Tip: When you use a TEMPLATE in the OUTDDN keyword with a SELECT FROM LIST (fast listdef select block), and when some of the table spaces that are in the list are multi-table table spaces, include a user-defined variable that is set to the :SELNUM predefined variable, using a TEMPLATESET option in the DSN expression of the TEMPLATE.

The OUTDDN keyword is required for SELECT statements.

MAX_EXPECTED_ROWS *n*

Specifies an estimate of the number of unloaded rows. Use this keyword when a WHERE clause is specified because the number of unloaded rows is less than the number of rows that are contained in the table. You can use MAX_EXPECTED_ROWS to do the following functions:

- Limit the number of rows that are specified to the sort utility.
- Limit the size of the allocated output dataset when you use a template without allocation parameters. When you specify MAX_EXPECTED_ROWS, data is unloaded by DB2 into a template file without space allocation being specified.

Consider specifying MAX_EXPECTED_ROWS if DB2 HPU fails because of a lack of resources. However, if the DB2 HPU estimate is accurate, specifying MAX_EXPECTED_ROWS will not solve resource problems.

Important: Specifying MAX_EXPECTED_ROWS disables the index scan feature and might decrease the DB2 HPU performance.

OUTMAXROWS *n*

Specifies the maximum number of rows to be extracted for this SELECT statement. If you are using DB2 HPU to natively process a partitioned table space by using partition parallelism, *n* applies to each partition.

Important: If you specify OUTMAXROWS, do not specify the SQLPART keyword because these keywords are incompatible with each other.

ON_RESULT_TABLE

Specifies that the maximum number of rows is applied when a SELECT statement is processed natively by DB2 HPU and when a SELECT statement, with or without SORT(EXTERNAL) specified, is processed by DB2. If an ORDER BY clause is specified, the

ON_RESULT_TABLE keyword is applied after the WHERE clause is applied and after the data is sorted.

For SELECT statements with SORT(INTERNAL) specified and that are processed by DB2, specifying ON_RESULT_TABLE has no effect because the setting corresponds with the default processing.

For SELECT statements with SORT(EXTERNAL) specified and that are processed by DB2, the OUTMAXROWS keyword is applied by default after data from the SQL cursor is fetched before the sort. If you specify ON_RESULT_TABLE, all data that is returned by the DB2 cursor is processed by the sort, and the OUTMAXROWS keyword allows the first *n* rows to be retrieved after sorting.

For SELECT statements that are processed natively by DB2 HPU, the OUTMAXROWS keyword is applied by default when data from the pages of the unloaded table space is read, before the WHERE clause is applied, and before the data is sorted, if an ORDER BY clause is specified. However, if you specify ON_RESULT_TABLE, the OUTMAXROWS keyword is applied after the WHERE clause is applied and after the sort by the ORDER BY clause.

Attention: Specifying ON_RESULT_TABLE might adversely affect DB2 HPU performance because a large amount of data might be processed before the number of records is limited.

OUTFREQROWS *n*

Specifies the unload sampling frequency.

If the SELECT statement is processed natively by DB2 HPU, one row in every *n* rows is kept when reading the rows from the linear data set (LDS) that contains the table space. In this case, the WHERE clause and the ORDER BY clause, if they exist, are applied after the sampling is done.

If the SELECT statement is processed by DB2, one row in every *n* rows is kept when fetching the rows from the cursor that is dynamically prepared by DB2 HPU for the complete SELECT statement, including the WHERE clause and the ORDER BY clause. In this case, if you specify SORT(INTERNAL), the WHERE clause and the ORDER BY clause, if they exist, are applied before the sampling is done. If the SORT(EXTERNAL) option is active, only the WHERE clause is applied before the sampling is done.

Important: If you specify OUTFREQROWS, do not specify the SQLPART keyword because these keywords are incompatible with each other.

ON_RESULT_TABLE

Specifies that the sampling for the SELECT statement is applied when a SELECT statement is processed natively by DB2 HPU or when a SELECT statement, with or without SORT(EXTERNAL) specified, is processed by DB2. The OUTFREQROWS keyword is applied after the WHERE clause is applied and after the data is sorted if an ORDER BY clause is specified.

For SELECT statements with SORT(INTERNAL) specified and that are processed by DB2, specifying ON_RESULT_TABLE has no effect because the setting corresponds with the default processing.

For SELECT statements with SORT(EXTERNAL) specified and that are processed by DB2, the OUTFREQROWS keyword is applied by default after data is fetched from the SQL cursor and before the sort. If you specify ON_RESULT_TABLE, all data that is returned by the DB2 cursor is processed by the sort, and the OUTFREQROWS keyword causes sampling after the sort.

For SELECT statements that are processed natively DB2 HPU, the OUTFREQROWS keyword is applied by default when data from the pages of the unloaded table space is read, before the WHERE clause is applied, and before the data is sorted by the ORDER BY clause. However, if you specify ON_RESULT_TABLE, the OUTFREQROWS keyword is applied after the WHERE clause is applied and after the sort by the ORDER BY clause.

Attention: Specifying ON_RESULT_TABLE might adversely affect DB2 HPU performance because a large amount of data might be processed before the records are sampled.

OUTEXIT *exit-name*

Specifies the name and the language of the exit that handles the rows during unload processing. The exit that you specify is loaded dynamically during unload processing. The exit must reside in an authorized library that must be in either the LINKLIST or an authorized JOBLIB or STEPLIB. For COBOL/2 and C, the STEPLIB, JOBLIB, or LINKLIST must also point to the LE/370 runtime libraries.

ASM Assembly language

C C language

COBOL2
COBOL/2 language

The default value is ASM.

EBCDIC/ASCII/UNICODE/ASIS

Specifies whether the data is unloaded in EBCDIC, ASCII, or UNICODE format by using the coded character set identifier (CCSID) of the installation or the specified CCSID.

ASCII Indicates that the unloaded data must be in ASCII format. DB2 HPU uses the ASCII CCSID of the subsystem, unless you override it by specifying the CCSID option.

ASIS Indicates that the data is unloaded in its original format. If the specification for the underlying table space cannot be determined, such as when the data is processed by DB2, the CCSID that is returned by a standard PREPARE statement that is used in an SQL descriptor area (SQLDA) is used. You can override ASIS by specifying the CCSID keyword.

Specifying ASIS does not mean that conversion is not required. Conversion might still be required in some situations, such as between input from SYSIN and the CCSID of the system, or between the CCSID of the system and printed output.

EBCDIC

Indicates that the data is unloaded in EBCDIC format. DB2 HPU uses the EBCDIC CCSID subsystem, unless you override it by specifying the CCSID keyword.

UNICODE

Indicates that the data is unloaded in UNICODE format. DB2 HPU uses the UNICODE CCSID of the subsystem, unless you override it by specifying the CCSID option.

The default value is EBCDIC.

Attention: If the unload format that was specified in either the SYSIN or in the PARMLIB by using the UNLSCHEM parameter is not identical to the EBCDIC format of the system, all constants that are specified in SYSIN are translated to the unload format.

Note: No translation is allowed for LOB/XML data when they are unloaded in VBS file (SPANNED YES option)

CCSID *integer*

Specifies up to three coded character set identifiers (CCSIDs) for the unloaded data. The first identifier specifies the CCSID for SBCS data, the second identifier specifies the CCSID for MIXED DBCS data, and the third identifier specifies the CCSID for DBCS data. If you omit any of these CCSIDs or specify 0 for any of them, the CCSID of the corresponding data type is assumed to be the same as the installation default CCSID.

You can also specify the CCSID at the column level in the USER block syntax.

The default value is 0.

Note: No translation is allowed for LOB/XML data when they are unloaded in VBS file (SPANNED YES option)

format block

See "FORMAT block syntax and description" on page 190.

options block

See "OPTIONS block syntax and description" on page 134.

See the *DB2 Universal Database for z/OS SQL Reference* for syntax and definitions for DB2 full select.

Related concepts:

Chapter 6, "DB2 HPU user exit," on page 257

The DB2 HPU user exit is used to customize the output data set that is created by a SELECT statement.

Related reference:

"User-allocated ddnames" on page 80

To run unload jobs, you must allocate certain ddnames in the DB2 HPU JCL.

"SELECT statement examples" on page 199

These SELECT statement examples show how to create sequential data sets in different formats, how to use the LISTDEF and TEMPLATE keywords, how to use the INTO clause and the REFORMAT clause, and how to use other DB2 HPU options.

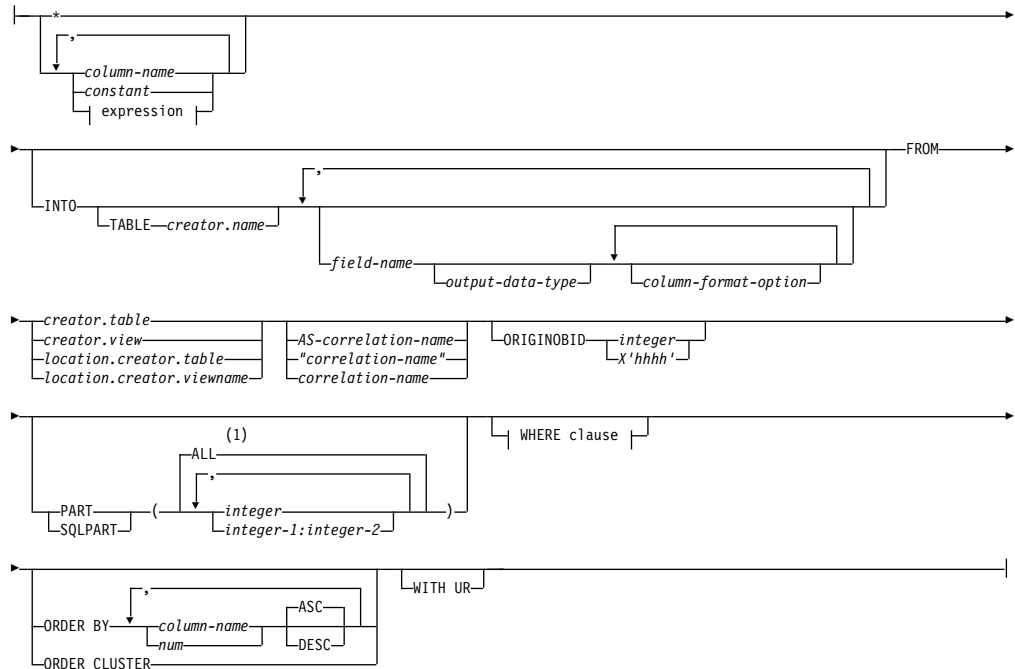
Fast select and fast listdef select blocks syntax and description

The fast select and fast listdef select blocks describe the syntax for SELECT statements that can be supported natively by DB2 HPU when DB2 NO or DB2 YES is specified in the UNLOAD command. The fast select and fast listdef select blocks are part of the SELECT block.

The INTO clause of the fast select block can also be used for any fullselect component that is coded in the DB2 HPU syntax even if the fullselect does not follow the syntax rules of the fast select block. To use the INTO clause for an unsupported SELECT statement, include the INTO clause in the SELECT statement between the list of selected items and the FROM clause. If the fullselect contains a UNION, EXCEPT, or INTERSECT clause, an INTO clause can be coded only on the first SELECT statement of the fullselect.

The following diagram shows the syntax of the fast select block:

fast select block:



fast listdef select block:



Notes:

- 1 If you do not specify the PART keyword or SQLPART keyword, the default value is PART(ALL).

The fast select block contains the following keywords. For the subset of keywords that the fast listdef select block accepts, see the syntax diagram.

* Indicates that this SELECT statement applies to all columns of the table.

column-name

Indicates the name of a column in the table.

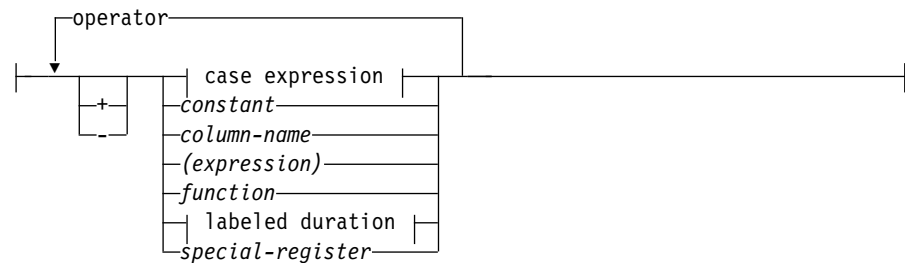
constant

Specifies a literal or a numeric value. If you specify a literal value, enclose it in single quotation marks (') or double quotation marks ("), according to your DB2 installation.

expression

The following syntax diagram shows an *expression*:

expression:



operator:



INTO

If the INTO clause is coded with the LIKE clause in FORMAT DSNTIAUL or FORMAT VARIABLE, field reformatting is not allowed, and you can use only the TABLE keyword.

You cannot use the INTO clause to rename or reformat the output fields, such as when you specify FORMAT USER.

TABLE *creator.name*

Specifies the name of the creator and the name of the target table in the RELOAD command. When you use the INTO TABLE *creator.name* clause together with the LIKE *creator.name* clause, the INTO TABLE *creator.name* clause provides the name of the table to be reloaded, and the LIKE *creator.name* clause provides the description of the columns for the RELOAD command.

field-name

Use this variable to rename the output fields and to define conversion and formatting options at the field level for all selected items. When you use this variable to change the data type and format of the output fields, the default format for the corresponding fields is overridden.

The description of the output fields in the INTO clause is a list that can contain one occurrence of each selected item. This list is positional, and the order corresponds to the order of the selected items in the SELECT statement. The list can be shorter than the list of selected items, but it cannot be longer. If the list is shorter, the unspecified selected items keep their default names and formats, which depend on the format that you specified. You do not need to specify the *field-name* and the description of the output field for each selected item. If you do not specify a field description for one selected item, you must use a comma (,) as a placeholder.

output-data-type

Changes the output data type.

column-format-option

Changes the formatting options of the output data.

FROM

Specifies the table or view from which columns are selected.

creator.table

Columns are selected from the table.

creator.view

Columns are selected from the view.

When you define SELECT statements that select data from views, the SELECT statements that you code must meet the following conditions if you want DB2 HPU to process the SELECT statement directly:

- The view can be defined only on a unique table. You cannot include joins, views of views, or subselects in the view definition.
- Each column of the view must precisely correspond to a column of the table. You cannot use scalar functions, column functions, expressions, or literals.
- The view cannot contain a WHERE clause.

DB2 HPU can process view definitions that do not conform to these conditions if DB2 YES or DB2 FORCE is specified. In these cases, the retrieval of the data is performed by DB2, but all other processing of the data is performed by DB2 HPU.

location.creator.table

location.creator.viewname

If the location that is used is identical to the current server (local DB2), DB2 HPU checks whether the SELECT statement can be processed. Otherwise, depending on whether you specified YES, NO, or FORCE for the DB2 parameter, the SELECT statement is not processed natively by DB2 HPU, and DB2 processing is attempted.

correlation-name

You can use a correlation name in a fast select block. If the AS keyword is used or if the correlation name is delimited with quotation marks, you can use any correlation name that is accepted by DB2. If any of the following keywords are used as a correlation name, they must be enclosed in quotation marks or preceded by the AS keyword:

- SELECT
- FROM
- ALL
- CHECK
- CURRENT
- END
- LIKE
- LOCK
- NO
- NULL
- WITH

- DATABASE
- TABLESPACE
- TABLE
- INDEX
- WHERE
- ORDER
- ORIGINOBID
- GROUP
- HAVING
- AS
- PART
- FORMAT

LIST(*list-name*)

Identifies the name of a list of objects that is defined by a LISTDEF control statement. The list can include table spaces, index spaces, databases, and partitions. The list cannot include LOB table spaces and directory objects. The list that is generated by the LISTDEF identifies the following objects:

- The tables from which the data is to be unloaded. You can use the pattern-matching feature of LISTDEF.
- For partitioned table spaces, the partitions from which the data will be unloaded. These partitions are defined by the INCLUDE, EXCLUDE, and PARTLEVEL keywords in the LISTDEF statement.

When you specify the LIST keyword in the fast listdef select block, DB2 HPU generates a SELECT statement for each table of each table space that is contained in the list. Using a TEMPLATE statement in the OUTDDN keyword is recommended. To activate partition parallelism with a TEMPLATE statement for partitioned tables, the data set name that is defined in the TEMPLATE statement must contain the &PART variable.

CLONE

Indicates that DB2 HPU is to unload data only from clone tables that are in the list of table spaces that are defined by the LISTDEF. If the list contains only table spaces without clones, no SELECT statement is generated.

When the CLONE keyword is not specified, DB2 HPU generates only SELECT statements on the base tables. The CLONE keyword can be specified only when you specify SELECT FROM LISTDEF. To unload data from a clone table by specifying SELECT FROM *table_name*, specify the name of the clone table in the FROM clause.

ORIGINOBID

Indicates when the OBID table in the image copy is not the same as the OBID that is read in the catalog. This situation can occur when an image copy of a table that is dropped and then re-created with a new OBID.

If the source data is an image copy, use this keyword to specify the OBID of the rows to be processed in this image copy.

integer

If the image copy file contains a unique table that you can use, you can specify 0 instead of the OBID of the table. If you specify the 0, DB2 HPU processes the first OBID that is found in the image copy.

X'hhhh'

X'hhhh' is the hexadecimal value of the OBID of the table in the image copy.

This keyword must be specified with the COPYDDN statement.

- PART** Specifies the partitions of the table space to be processed. You can specify the PART keyword in the UNLOAD block and the SELECT block.
- If PART is not specified in the UNLOAD block, then PART(ALL) is the default.
 - If PART is not specified in a SELECT block of this UNLOAD block, the PART specification in the UNLOAD block is used.
 - If you specify PART in any SELECT statement, it overrides any values that are specified in the UNLOAD block.
 - For each UNLOAD statement, the union of all the subsets of partitions that are selected in all SELECT blocks must be equal to the subset that is specified in the UNLOAD PART, unless UNLDDN is used.
 - If a statement is processed by DB2, PART is ignored.

Requirement: If you are unloading from partition-level full image copies, you must provide a uniquely named DD statement for each partition to be unloaded. For example, if one SELECT statement unloads from partitions 1, 2, and 4, and another SELECT statement unloads from partitions 2 and 3, you must specify a DD statement with a unique name for image copy partitions 1, 2, 3, and 4.

integer

Indicates which partitions are to be processed.

ALL Specifies that the entire table space is to be processed.

integer-1:integer-2

Designates a range of partitions *integer-1 -integer-2*. *Integer-1* must be positive and less than the highest partition number in the table space. *Integer-2* must be greater than *Integer-1* and less than or equal to the highest partition number.

The default value is ALL.

Attention: Do not use this option when the table space is simple or segmented.

SQLPART

Specifies the partitions of the table space to be processed. The SQLPART parameter applies to a physical partition number, which can be different from logical partitions. Unlike the PART keyword, you can specify SQLPART only in the SELECT block. SQLPART is used only when partitioned table spaces are unloaded.

SQLPART applies to SELECT statements that are processed natively by DB2 HPU and to SELECT statements that are processed by DB2. SQLPART is not accepted when either the LISTDEF option or the DDLDDN option is used.

SQLPART is used only on table spaces that are partitioned by range or partitioned by index. Table spaces that are partitioned by growth or that are not partitioned are not supported.

SQLPART acts according to the following specifications in the UNLOAD command:

- If DB2 NO is specified, SQLPART is equivalent to PART.
- If DB2 FORCE is specified, SQLPART allows unloading per partition. In this case, a separate UNLOAD file can be used for each partition, and parallelism is activated between the unloaded partitions.

- If DB2 YES is specified, SQLPART is processed as PART for supported SELECT statements. For unsupported SELECT statements, SQLPART allows unloading per partition.

If SQLPART is used and you are unloading partitioned table spaces, the rules for correspondence between partitions and output files apply.

Specifying SQLPART(ALL) is different from not specifying SQLPART in the following ways:

- When SQLPART is not specified, data is entirely unloaded into a single file. If a TEMPLATE is used to create the UNLOAD file, the &PART variable in the TEMPLATE is replaced with '00000'.
- When SQLPART(ALL) is specified, the processing depends on whether one output file per partition is used. If a TEMPLATE that contains the &PART variable is used, DB2 HPU allocates one output file for each partition, and data is unloaded on a per-partition basis. If JCL-allocated files are used, the type of process depends on whether files per partition are used.

SQLPART is incompatible with the OUTFREQROWS keyword and the OUTMAXROWS keyword.

When SQLPART is used, DB2 HPU determines the name of the unloaded table. Joins and unions are not allowed on the processed SELECT statement.

If the limit key for FLOAT and DECFLOAT columns is not defined explicitly when the partitioning key is created, DB2 HPU does not support using SQLPART with key limits for FLOAT and DECFLOAT columns of the partitioning key.

The following examples show how the SQLPART option is used:

Example: Obtaining separate files per partition with SQLPART in DB2 FORCE

The following example shows an unloaded table space that contains six partitions. The execution JCL contains two DD statements, SYSREC, and SYSREC2, which are allocated to two output files.

```
UNLOAD TABLESPACE DBDM8A.TSTEST16
QUIESCE NO LOCK YES QUIESCECAT YES
DB2 FORCE
SELECT * FROM TSTEST16 SQLPART(ALL)
OUTDDN(SYSREC)
FORMAT DSNTIAUL LOADDN SYSPUNCH
```

Data from partition 1 and partitions 3-6 is unloaded in the file SYSREC, and data from partition 2 is unloaded in file SYSREC2. At least two SELECT statements are used internally for the following partitions:

- One SELECT statement is used for partition 2, which writes in file SYSREC2 (second message INZR3000).
- One to five SELECT statements are used for partition 1 and partitions 3-6, which write in file SYSREC (third message INZR3000). The effective number of SELECT statements depends on correspondence between the physical partitions and the logical partitions. If physical partitions 3-6 correspond to four consecutive logical partitions, a single SELECT statement is used.

Otherwise, DB2 HPU generates a SELECT statement for each range of adjacent logical partitions.

The following SYSPRINT corresponds to partitioned table space DBDM8A.TSTEST16:

```
UDBU281I - UNLOAD STARTING AT POS(2, 1)

UDBU277I - PROCESSING UNLOAD 00001 FROM TABLESPACE DBDM8A.TSTEST16
UDBU360I - SELECT STATEMENT PROCESSING THROUGH DB2
UDBU280I - SELECT 00001 STARTING AT POS(6, 2)
UDBU282I OUTPUT DDNAME=SYSREC
INZR3000 NUMBER OF ROWS EXTRACTED COPIED ON SYSREC : 8333
INZR3000 NUMBER OF ROWS EXTRACTED COPIED ON SYSREC2 : 8333
INZR3000 NUMBER OF ROWS EXTRACTED COPIED ON SYSREC : 33334
UDBU260I GENERATING LOAD STATEMENT FOR SELECT STARTING AT POS(6, 2)
UDBU222I SYSREC , TOTAL NUMBER OF RECORDS WRITTEN 41667
UDBU376I SELECT 1 NUMBER OF RECORDS WRITTEN 41667
UDBU222I SYSREC2 , TOTAL NUMBER OF RECORDS WRITTEN 8333
UDBU376I SELECT 1 PARTITION 2 NUMBER OF RECORDS WRITTEN 8333
```

Example: Implicit usage of SORT EXTERNAL with SQLPART

When an ORDER BY clause is coded on the SELECT statement, DB2 HPU can switch automatically to SORT(EXTERNAL) processing to be able to sort data that comes from different ranges of partitions that are unloaded in a single output file. The following example shows an implicit usage of SORT(EXTERNAL). A single SYSREC file is allocated in the JCL.

```
//UNLOAD EXEC PGM=INZUTILK,PARM='DZ8A,INFOUNLOAD',
// REGION=0M,TIME=1440
//STEPLIB DD DSN=HINZ320.SINZLINK,DISP=SHR
// DD DSN=DSN810.SDSNLOAD,DISP=SHR
//SYSREC DD DSN=...
//UTPRINT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSPUNCH DD SYSOUT=*
//SYSIN DD *
UNLOAD TABLESPACE DBDM8A.TSTEST16
QUIESCE NO LOCK YES QUIESCECAT YES
DB2 FORCE
SELECT * FROM TSTEST16 SQLPART(1:3,5:6)
ORDER BY COLINTEGER
OUTDDN(SYSREC)
FORMAT DSNTIAUL LOADDDN SYSPUNCH
//
```

In this example, two SELECT statements are generated to process the two ranges of partitions. Data from partitions 1 - 3 might be mixed with data from partitions 5 - 6 because both SELECT statements are processed in parallel. For these two ranges of partitions, DB2 HPU processes the sort that corresponds to the ORDER BY specification based on the result of merging the two SELECT statements to obtain data that is written in file SYSREC and sorted on column COLINTEGER.

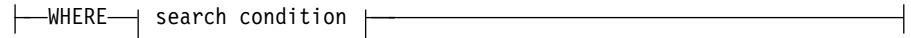
When the SORT(EXTERNAL) option is specified in DB2 HPU, and the SORT specification is not supported natively, the sort is not done, warning message UDBU511W is issued, and the output data is not sorted.

When an ORDER BY clause is coded on the SELECT statement, DB2 HPU can automatically switch to SORT(EXTERNAL) processing to be able to sort data that comes from different ranges of partitions that are unloaded in a single output file.

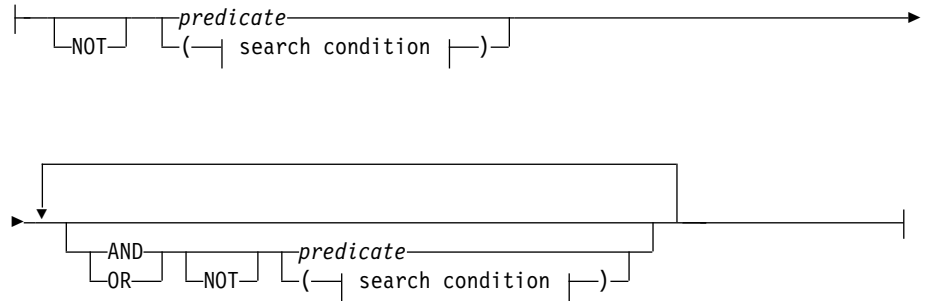
WHERE

The following syntax diagram shows the format of the WHERE clause:

WHERE clause:



search condition:



search condition

NOT

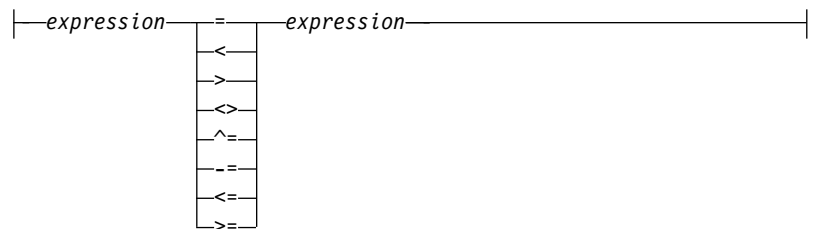
predicate

The following predicates specify a comparison between two expressions.

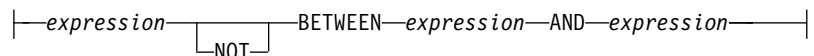
- basic predicate
- between predicate
- in predicate
- like predicate
- null predicate

The following syntax diagram shows the predicate syntax:

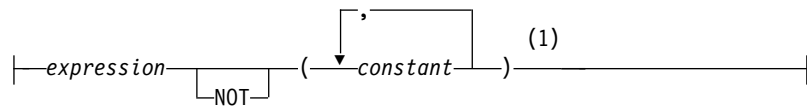
basic predicate:



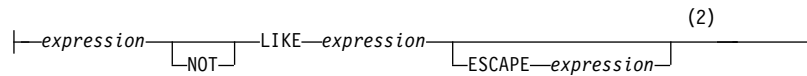
between predicate:



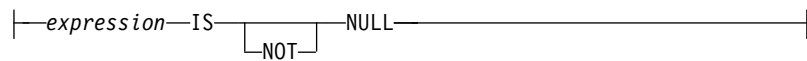
in predicate:



like predicate:



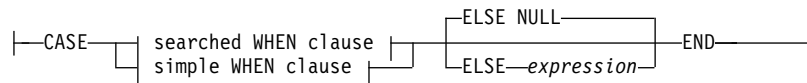
null predicate:



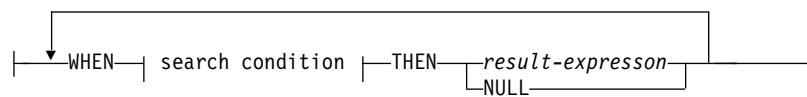
Notes:

- 1 The IN predicate compares an expression with a set of values. The result of this predicate is true if the value of the expression matches one in the list; otherwise, the result is false.
- 2 The LIKE predicate works the same as the standard SQL LIKE predicate.

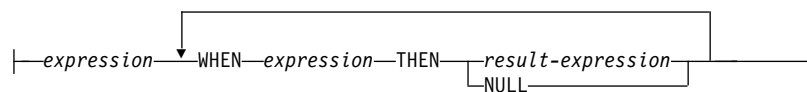
case expression:



searched WHEN clause:



simple WHEN clause:



labeled duration:

—constant	YEAR	-----
	YEARS	
	MONTH	
	MONTHS	
	DAY	
	DAYS	
	HOUR	
	HOURS	
	MINUTE	
	MINUTES	
	SECOND	
	SECONDS	
	MICROSECOND	
	MICROSECONDS	

expression

An SQL expression is a combination of columns, functions and special registers and operators.

The tables below give the list of supported and unsupported SQL items (scalar functions and operators). Be aware that SQL statements containing only supported items may not be supported as some combinations may prevent DB2 HPU to process the statement in native mode.

The following table contains a list of supported and unsupported SQL:

Table 27. SQL supported and unsupported list

Items	Supported? (1)
ABS	SUPPORTED
ADD_MONTHS	UNSUPPORTED (3)
AND	SUPPORTED
ANY	SUPPORTED
ASCII	UNSUPPORTED (4)
ASCII_CHR	UNSUPPORTED (4)
ASCII_STR	UNSUPPORTED (4)
ASIN	UNSUPPORTED (4)
ATAN	UNSUPPORTED (4)
BETWEEN	SUPPORTED
BIGINT	SUPPORTED
BINARY	UNSUPPORTED (4)
BITAND	UNSUPPORTED (3)
BITANDNOT	UNSUPPORTED (3)
BITNOT	UNSUPPORTED (3)
BITOR	UNSUPPORTED (3)
BITXOR	UNSUPPORTED (3)
BLOB	UNSUPPORTED (4)
CASE expression	SUPPORTED
CCSID_ENCODING	UNSUPPORTED (4)

Table 27. SQL supported and unsupported list (continued)

Items	Supported? (1)
CEIL	SUPPORTED
CEILING	SUPPORTED
CHAR	SUPPORTED
CHARACTER_LENGTH	UNSUPPORTED (4)
CLOB	UNSUPPORTED (4)
COALESCE	SUPPORTED
COLLATION_KEY	UNSUPPORTED (4)
COMPARE_DECFLOAT	UNSUPPORTED (3)
CONCAT	SUPPORTED
CONTAINS	UNSUPPORTED (4)
COS	UNSUPPORTED (4)
COSH	UNSUPPORTED (4)
CURRENT	SUPPORTED TIME REGISTER
CURRENT_TIME	SUPPORTED TIME REGISTER
CURRENT_TIMESTAMP	SUPPORTED TIME REGISTER
DATE	SUPPORTED
DAY	SUPPORTED
DAYOFMONTH	SUPPORTED
DAYOFWEEK	SUPPORTED
DAYOFWEEK_ISO	UNSUPPORTED (4)
DAYOFYEAR	SUPPORTED
DAYS	SUPPORTED
DBCLOB	UNSUPPORTED (4)
DEC	SUPPORTED
DECFLOAT	SUPPORTED
DECFLOAT_FORMAT	UNSUPPORTED (4)
DECFLOAT_SORTKEYS	UNSUPPORTED (4)
DECIMAL	SUPPORTED
DECRYPT_BINARY	UNSUPPORTED (4)
DECRYPT_BIT	UNSUPPORTED (4)
DECRYPT_CHAR	UNSUPPORTED (4)
DECRYPT_DB	UNSUPPORTED (4)
DEGREES	UNSUPPORTED (3)
DIFFERENCE	UNSUPPORTED (4)
DIGITS	SUPPORTED
DOUBLE	SUPPORTED
DOUBLE_PRECISION	SUPPORTED
DSN_XMLVALIDATE	UNSUPPORTED (4)

Table 27. SQL supported and unsupported list (continued)

Items	Supported? (1)
EBCDIC_CHAR	UNSUPPORTED (4)
EBCDIC_STR	UNSUPPORTED (4)
ENCRYPT_TDES	UNSUPPORTED (4)
EXP	UNSUPPORTED (3)
EXTRACT	UNSUPPORTED (4)
FLOAT	SUPPORTED
FLOOR	SUPPORTED
GENERATE_UNIQUE	UNSUPPORTED (4)
GETHINT	UNSUPPORTED (4)
GETVARIABLE	SUPPORTED
GRAPHIC	SUPPORTED
GREATEST	UNSUPPORTED (3)
HEX	SUPPORTED
HOUR	SUPPORTED
HOURS	SUPPORTED
IDENTIFY_VAL_LOCAL	UNSUPPORTED (4)
IF statement	SUPPORTED
IFNULL	SUPPORTED
IN	SUPPORTED
INSERT	UNSUPPORTED (4)
INT	SUPPORTED
INTEGER	SUPPORTED
IS	SUPPORTED
JULIAN_DAY	UNSUPPORTED (4)
LAST_DAY	UNSUPPORTED (3)
LCASE	UNSUPPORTED (4)
LEAST	UNSUPPORTED (3)
LEFT	UNSUPPORTED (2)
LENGTH	SUPPORTED
LIKE	SUPPORTED
LN	UNSUPPORTED (3)
LOCAL	SUPPORTED
LOCATE	UNSUPPORTED (4)
LOG	UNSUPPORTED (3)
LOG10	UNSUPPORTED (3)
LOWER	UNSUPPORTED (4)
LPAD	UNSUPPORTED (3)
LTRIM	SUPPORTED
MAX	UNSUPPORTED (3)
MICROSECOND	SUPPORTED

Table 27. SQL supported and unsupported list (continued)

Items	Supported? (1)
MICROSECONDS	SUPPORTED
MIDNIGHT_SECONDS	SUPPORTED
MIN	SUPPORTED
MINUTE	SUPPORTED
MINUTES	SUPPORTED
MOD	SUPPORTED
MONTH	SUPPORTED
MONTH_BETWEEN	UNSUPPORTED (4)
MONTHS	SUPPORTED
MONTHS_BETWEEN	UNSUPPORTED (3)
MQPUBLISH	UNSUPPORTED (4)
MQPUBLISHXML	UNSUPPORTED (4)
MQREAD	UNSUPPORTED (4)
MQREADCLOB	UNSUPPORTED (4)
MQREADXML	UNSUPPORTED (4)
MQRECEIVE	UNSUPPORTED (4)
MQRECEIVELOB	UNSUPPORTED (4)
MQRECEIVEXML	UNSUPPORTED (4)
MQSEND	UNSUPPORTED (4)
MQSENDXML	UNSUPPORTED (4)
MQSENDXMLFILE	UNSUPPORTED (4)
MQSENDXMLFILECLOB	UNSUPPORTED (4)
MQSUBSCRIBE	UNSUPPORTED (4)
MQUNSUSBSCRIBE	UNSUPPORTED (4)
MULTIPLY_ALT	UNSUPPORTED (4)
NEXT_DAY	UNSUPPORTED (4)
NORMALIZE_DECFLOAT	UNSUPPORTED (3)
NORMALIZE_STRING	UNSUPPORTED (4)
NOT	SUPPORTED
NULLIF	SUPPORTED
NVL	SUPPORTED
OR	SUPPORTED
OVERPLAY	UNSUPPORTED (4)
POSITION	UNSUPPORTED (4)
POSSTR	UNSUPPORTED (4)
POWER	UNSUPPORTED (3)
QUANTIZE	UNSUPPORTED (3)
QUARTER	SUPPORTED
RADIANS	UNSUPPORTED (4)
RAISE_ERROR	UNSUPPORTED (4)

Table 27. SQL supported and unsupported list (continued)

Items	Supported? (1)
RAND	UNSUPPORTED (4)
REAL	SUPPORTED
REPEAT	UNSUPPORTED (3)
RID	UNSUPPORTED (4)
RIGHT	UNSUPPORTED (2)
ROUND	UNSUPPORTED (4)
ROUND_TIMESTAMP	UNSUPPORTED (4)
ROWID	UNSUPPORTED (4)
RPAD	UNSUPPORTED (3)
RTRIM	SUPPORTED
SCORE	UNSUPPORTED (4)
SECOND	SUPPORTED
SECONDS	SUPPORTED
SIGN	SUPPORTED
SIN	UNSUPPORTED (4)
SINH	UNSUPPORTED (4)
SMALLINT	SUPPORTED
SOAPHTTPC	UNSUPPORTED (4)
SOAPHTTPNC	UNSUPPORTED (4)
SOAPHTTPNV	UNSUPPORTED (4)
SOAPHTTPV	UNSUPPORTED (4)
SOUNDEX	UNSUPPORTED (4)
SPACE	UNSUPPORTED (4)
SQRT	UNSUPPORTED (3)
STRIP	SUPPORTED
SUBSTR	SUPPORTED
TAN	UNSUPPORTED (4)
TANH	UNSUPPORTED (4)
TIME	SUPPORTED
TIMESTAMP	SUPPORTED
TIMESTAMP_FORMAT	UNSUPPORTED (4)
TIMESTAMP_iso	UNSUPPORTED (3)
TIMESTAMPADD	UNSUPPORTED (4)
TIMESTAMPDIFF	UNSUPPORTED (3)
TO_NUMBER	UNSUPPORTED (3)
TOTALORDER	UNSUPPORTED (4)
TRANSLATE	UNSUPPORTED (4)
TRIM	SUPPORTED
TRUNC	SUPPORTED
TRUNC_TIMESTAMP	UNSUPPORTED (4)

Table 27. SQL supported and unsupported list (continued)

Items	Supported? (1)
TRUNCATE	SUPPORTED
UCASE	UNSUPPORTED (4)
UNICODE	UNSUPPORTED (4)
UNICODE_STR	UNSUPPORTED (4)
UPPER	UNSUPPORTED (4)
VALUE	SUPPORTED
VARBINARY	UNSUPPORTED (4)
VARCHAR	SUPPORTED
VARCHAR_FORMAT	UNSUPPORTED (4)
VARGRAPHIC	SUPPORTED
VERIFY_GROUP_FOR_USER	UNSUPPORTED (4)
VERIFY_ROLE_FOR_USER	UNSUPPORTED (4)
VERIFY_TRUSTED_CONTECT_ROLE_FOR_USER	UNSUPPORTED (4)
WEEK	UNSUPPORTED (4)
WEEK_ISO	UNSUPPORTED (4)
XMLATTRIBUTES	UNSUPPORTED (4)
XMLCOMMENT	UNSUPPORTED (4)
XMLCONCAT	UNSUPPORTED (4)
XMLDOCUMENT	UNSUPPORTED (4)
XMLELEMENT	UNSUPPORTED (4)
XMLFOREST	UNSUPPORTED (4)
XMLNAMESPACES	UNSUPPORTED (4)
XMLPARSE	UNSUPPORTED (4)
XMLPI	UNSUPPORTED (4)
XMLQUERY	UNSUPPORTED (4)
XMLSERIALIZE	UNSUPPORTED (4)
YEAR	SUPPORTED
YEARS	SUPPORTED

Notes:

- SUPPORTED? (1): The column represents the list of scalars and words known or unknown by DB2 HPU
- SUPPORTED: Scalar function known by DB2 HPU.
- UNSUPPORTED (2): DB2 HPU only qualifies the whole SQL statements as unsupported but not specific error message is issued for them.
- UNSUPPORTED (3): DB2 HPU issues a specific error message mentioning this item. The location is not mentioned within the SQL statement.
- UNSUPPORTED (4): DB2 HPU issues the INZU228E message and indicates the name and the location of this unsupported item.

ORDER BY

Indicates that the output data set must be sorted based on the chosen criteria.

ASC Specifies that the rows are sorted in ascending order and the column name or an integer that identifies the *n*th column of the result table.

DESC Specifies that the rows are sorted in descending order and the column name or an integer that identifies the *n*th column of the result table.

If the ORDER BY clause corresponds to the order of the clustering index, DB2 HPU tries to use this index to unload the rows in the required order. In all other cases, a SORT is issued.

The default value is ASC.

ORDER CLUSTER

Indicates that the output data set must be sorted according to the index cluster. To display the sort messages, code the UTPRINT ddname. If an index is not defined on the table, a warning message is issued and processing continues.

WITH UR

This clause is accepted for SELECT statements that are processed natively only for compatibility with SELECT statements that are processed by using DB2. Internal lock mechanisms do not apply because the SELECT statements that are processed natively are processed outside of DB2. Specifying the WITH UR clause does not affect the results of SELECT statements that are processed natively and does not affect the integrity of data that is unloaded by DB2 HPU.

The LOCK and QUIESCE options apply to the SELECT statements that are processed natively or processed by using DB2 even if the WITH UR clause is specified.

Restriction: The WITH UR clause is not supported in the fast listdef select block.

The following example shows how you can unload data with the WITH UR clause in native mode:

```
UNLOAD TABLESPACE
QUIESCE YES
DB2 NO
SELECT * FROM DSN8810.EMP
WITH UR
FORMAT DSNTIAUL
OUTDDN(OUTPUT)
```

See the *DB2 Universal Database for z/OS SQL Reference* for more information about function parameters.

Related concepts:

“Ddname allocation” on page 79

DB2 HPU uses two types of ddnames: ddnames that DB2 HPU allocates and ddnames that you must allocate.

Related reference:

“OPTIONS block syntax and description” on page 134

Use the OPTIONS block to specify the default conversions that are with the SELECT statements. This block can be used in the GLOBAL block, the UNLOAD

block, and the SELECT block.

Example: Unloading data from a table with informational header:

This example shows how to unload data from a table with a line header saying from which DB2 subsystem the data has been unloaded.

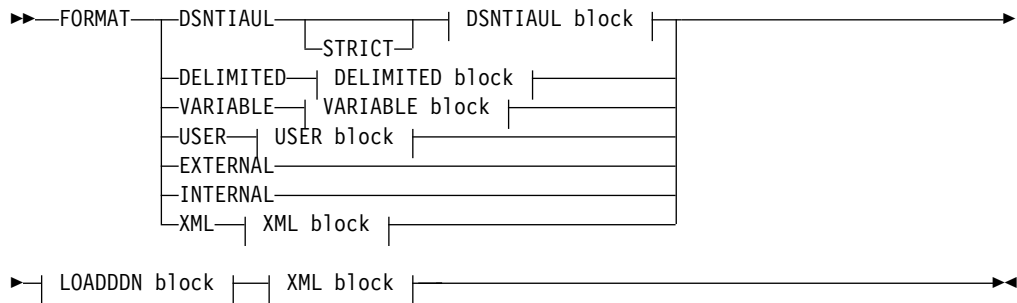
```
UNLOAD TABLESPACE
DB2 NO
SELECT CHAR(
    CHAR('SOURCE_SSID='),!!
    CHAR(GETVARIABLE('SYSIBM.SYSTEM_NAME')),
    16),
    C1,
    C2,
    C3
FROM MZLFLB.TBDZ9AFLB00101
OUTDDN(SYSRECC01)
FORMAT DELIMITED
```

FORMAT block syntax and description

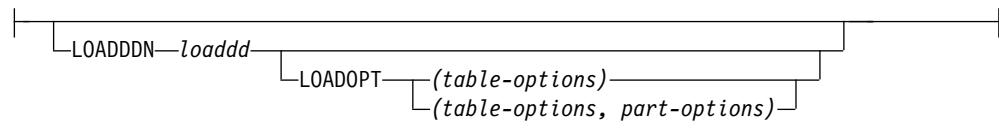
Use the FORMAT block to specify the format of the data that is unloaded. The FORMAT block is a part of the SELECT block.

The following diagram shows the syntax of the FORMAT block:

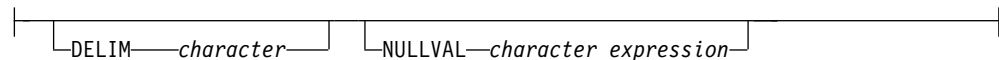
FORMAT block



LOADDDN block:



XML block:



You can specify the following keywords and options in the FORMAT block:

FORMAT

Indicates that this block is a FORMAT block. Use the FORMAT keyword to specify the format of the data that is unloaded.

DSNTIAUL dsntiaul block

Unloads data in the same format that is produced by the DSNTIAUL program.

Important: Depending on the content of the VUU057/OPALLFMT parameter, keywords and options that you specify in a GLOBAL block or UNLOAD block apply only to the USER format except for the LOADOPT, FLOAT, and UNLROWSET keywords. The LOADOPT and UNLROWSET keywords apply to all formats. The FLOAT keyword applies to DSNTIAUL, USER, and VARIABLE formats.

DELIMITED delimited block

Unloads data in DELIMITED format.

Important: Depending on the content of the VUU057/OPALLFMT parameter, options that you specify in a GLOBAL block or UNLOAD block apply only to the USER format except for the LOADOPT, FLOAT, and UNLROWSET keywords. The LOADOPT and UNLROWSET keywords apply to all formats. The FLOAT keyword applies to DSNTIAUL, USER, and VARIABLE formats.

VARIABLE variable block

Unloads data in a format that is compatible with the DB2 LOAD data set.

Important: Depending on the content of the VUU057/OPALLFMT parameter, options that you specify in a GLOBAL block or UNLOAD block apply only to the USER format except for the LOADOPT, FLOAT, and UNLROWSET keywords. The LOADOPT and UNLROWSET keywords apply to all formats. The FLOAT keyword applies to DSNTIAUL, USER, and VARIABLE formats.

USER user block

Unloads data in the defined format. You can specify the format of a specific column using a USER block.

EXTERNAL

Unloads data in EXTERNAL format. The EXTERNAL keyword contains the following characteristics:

- Output records are fixed by default.
- Variable-length columns that are at the maximum length are preceded with the two length bytes and are padded on the right with binary zeros.
- Nullable fields are followed by a byte that contains '00' if the data is not null and contains '?' if the data is null.
- A field separator is not used.

All fields are in the EXTERNAL format that corresponds to their default type.

INTERNAL

Unloads data in the DB2 INTERNAL format. This format is compatible with the INTERNAL format in the DB2 LOAD utility. The INTERNAL keyword is limited by the following restrictions:

- The INTERNAL keyword is accepted only on natively processed SELECT statements.
- The INTERNAL keyword is allowed only on a SELECT * FROM *table-name* statement or a SELECT FROM LIST(*listdef-name*) statement.

- The INTERNAL keyword is not accepted on views.
- The INTERNAL keyword cannot be combined with other format types in SELECT statements that are processed on the same table space with the same UNLOAD command.
- The INTERNAL keyword is not supported with tables that contain a LOB column or an XML column.
- When you use the INTERNAL keyword, an OPTIONS block that is used to change the formatting of the output record or a REFORMAT clause is ignored.
- If you specify the INTERNAL keyword, the OUTEXIT and CCSID keywords or encoding scheme specifications are ignored.
- You cannot specify the INTERNAL keyword with the OUTMAXROWS or OUTFREQROWS keywords.
- The INTO clause is not allowed for SELECT statements that use the INTERNAL keyword.
- The ORDER CLUSTER and ORDER BY clauses are accepted with the INTERNAL keyword in the following conditions:
 - The table is in reordered row format (RRF).
 - The table is in basic row format (BRF), and the ORDER CLUSTER clause or the ORDER BY clause does not include columns that are located after the first variable-length column in the table.
- When the INTERNAL keyword is specified, the setting of the VUU050/ULFRQMSG parameter is ignored, and information messages that display the number of currently written records are not displayed.

The following example shows how you can request a logical unload of data by using the INTERNAL keyword. Data from the DSN8810.EMP table is unloaded in the INTERNAL format.

```
UNLOAD TABLESPACE
QUIESCE YES
DB2 NO
SELECT * FROM DSN8810.EMP
FORMAT INTERNAL
OUTDDN(OUTPUT)
```

LOADDDN

Specifies the name of the DD statement that describes the command data set. Use this keyword if you want DB2 HPU to create a command data set for the DB2 LOAD utility. The corresponding DD statement must be present in the execution JCL. This data set contains the required commands for loading a sequential data set by using the DB2 LOAD utility.

If the LIKE *table-name* clause is not used, the model table is the table that is referred to in the SELECT statement.

LOADOPT

For a description of this keyword, see “OPTIONS block syntax and description” on page 134.

The description of the XML keyword is the following one:

DELIM delimiter character

Specifies the delimiter character to be used to enclose CHAR, VARCHAR, GRAPHIC and VARGRAPHIC fields in the output data set.

The value of the delimiter-character can be 'c' or X'hh'

DELIM does not have a default value. The value are not delimited if no DELIM option is specified.

NULLVAL character expression

Specifies the delimiter character string to be unloaded when a NULL value is to be unloaded. This character string is not delimited even if the DELIM option is specified.

If NULLVAL is not specified, the NULL value are not substituted and an empty string with no delimiter is unloaded.

Related reference:

“OPTIONS block syntax and description” on page 134

Use the OPTIONS block to specify the default conversions that are with the SELECT statements. This block can be used in the GLOBAL block, the UNLOAD block, and the SELECT block.

“DSNTIAUL block syntax and description”

Use the DSNTIAUL block to put the output data set in the same format that is produced by the DSNTIAUL program.

“DELIMITED block syntax and description” on page 195

Use the DELIMITED block to indicate that the format of the output data set is a comma-separated-value file, which corresponds to a .csv file that can be processed by your personal computer.

“VARIABLE block syntax and description” on page 196

Use the VARIABLE block to indicate that the output data set must be compatible with the DB2 LOAD data set.

“USER block syntax and description” on page 198

Use the USER block to indicate that the unloaded data is formatted according to the keywords that are specified in the USER block.

“DB2 HPU output data parameters” on page 410

The DB2 HPU output data parameters section on the Product Parameters panel (CCQPPRD) in Tools Customizer contains the parameters for configuring output data.

DSNTIAUL block syntax and description:

Use the DSNTIAUL block to put the output data set in the same format that is produced by the DSNTIAUL program.

The DATE and TIME columns are unloaded based on DB2 installation parameters.

The default data set record format is fixed blocked (FB), but it can be specified as fixed (F), variable (V), or variable blocked (VB) in the JCL.

If the LRECL is specified, it is kept by DB2 HPU. In this case, data can be truncated.

If the LRECL is not specified, DB2 HPU determines it at run time based on the following rules:

- If you specified RECFM=F, the LRECL of the data set is equal to the sum of the maximum length of fields, regardless of the LRECL value that you specified in the JCL. The output data set is in FB format.
- If you specified RECFM=V or RECFM=VB, the LRECL of the data set is equal to the sum of the maximum length of fields plus 4 bytes, regardless of the LRECL value that you specified in the JCL. The output data set is in VB format.

The length of variable-length fields is the sum of their maximum length plus 2 bytes.

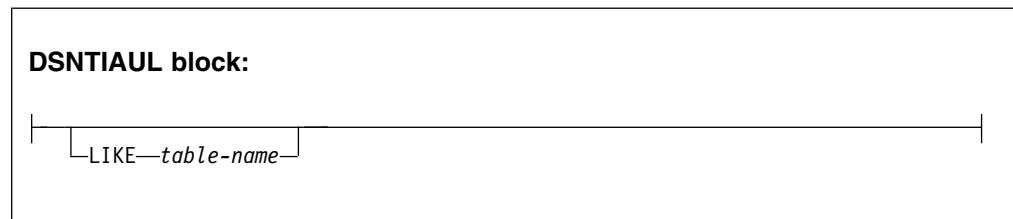
DSNTIAUL STRICT affects the formatting of constant character strings that are specified in SELECT statements. For example:

```
SELECT 'character-string'
```

If you specify DSNTIAUL, *character-string* is written to the output data set as type CHAR.

If you specify DSNTIAUL STRICT, *character-string* is written to the output data set as type VARCHAR.

The following syntax diagram shows the DSNTIAUL block:



If the LIKE *table-name* clause is not specified, the characteristics and sequence of fields in the generated data set are the same as the selected columns. The format of data that is written in the data set is in the same format that is generated by the DSNTIAUL program:

- Numeric data has a standard format.
- Columns of type DATE, TIME, or TIMESTAMP have the default format for the site.
- Variable-length fields (VARCHAR, LONG VARCHAR, VARGRAPHIC, and LONG VARGRAPHIC) contain length bytes at the beginning of the field. The field is extended to the maximum size of the column.
- If a column accepts nulls, a null-byte indicator is generated into the encoding scheme (EBCDIC, ASCII, or UNICODE) and into the CCSID of the output data. This indicator contains the value ? when the field is null, or a binary zero (x'00') if the value is not null. The hexadecimal value of ? depends on the encoding scheme.

LIKE *table-name*

If the LIKE *table-name* clause is specified, DB2 HPU uses the characteristics of the table model as parameters and formats the data set to allow this table to be loaded.

The sequence of columns in the SELECT statement must match the columns in the model table. If format conversions are required, they follow DB2 HPU format rules.

The *table-name* must be in the format *user_ID*.TABLE. If the user ID is not specified, the name of the user (CURRENT SQLID) is used.

Field types are provided by the table model. A null-byte indicator is generated behind the field if the column of the table model can be null.

If a column in the SELECT statement is null and if the corresponding column in the table model does not allow nulls, the field in the output sequential data set has one of the following standard default values:

- 0 for numeric fields (INTEGER, SMALLINT, DECIMAL, FLOAT)
- A string of blank characters (CHAR, GRAPHIC)
- Two bytes of zero length (VARCHAR, LONG VARCHAR, VARGRAPHIC, LONG VARGRAPHIC)
- 0001-01-01 (DATE)
- 00.00.00 (TIME)
- 0001-01-01-00.00.00.000000 (TIMESTAMP)

You can override the default DATE, TIME, and TIMESTAMP formats by specifying an OPTIONS block at the SELECT level. Only an OPTIONS block at the SELECT level is considered for this format.

Related reference:

“VARIABLE block syntax and description” on page 196

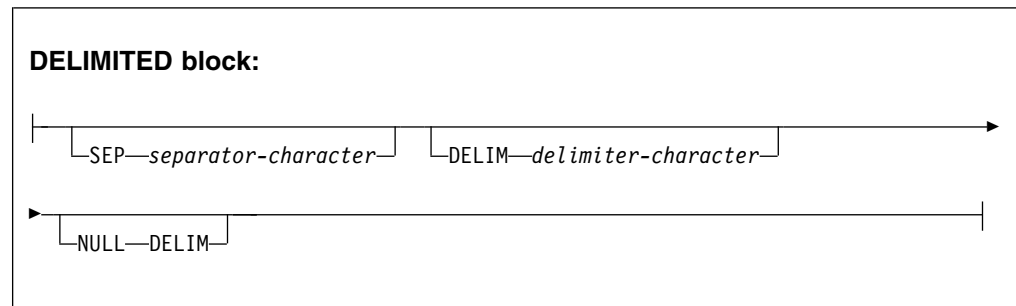
Use the VARIABLE block to indicate that the output data set must be compatible with the DB2 LOAD data set.

DELIMITED block syntax and description:

Use the DELIMITED block to indicate that the format of the output data set is a comma-separated-value file, which corresponds to a .csv file that can be processed by your personal computer.

When numeric and date, time, and timestamp data is unloaded in DELIMITED format, the data is always unloaded in an external, displayable format, even if an INTO clause or a REFORMAT clause specifies a conversion to INTERNAL format. For example, specifying REFORMAT (TYPE SMALLINT INTO DECIMAL(5,0)) generates a field that contains decimal external data. When data is converted into date, time, or timestamp data by using an INTO clause or a REFORMAT clause, the default output format is the ISO representation of this data. For example, specifying COL_DATE INTO C1 DATE returns a field in the ISO representation of a date column.

The following syntax diagram shows the format of the DELIMITED block:



SEP separator-character

Specifies the separator character to be used to separate fields in the output data set.

The value of the *separator-character* can be 'c' or X'hh'.

The default value is (X'40').

DELIM delimiter-character

Specifies the delimiter character to be used to enclose CHAR, VARCHAR, GRAPHIC, and VARGRAPHIC fields in the output data set.

The value of the *delimiter-character* can be 'c' or X'hh'.

DELIM does not have a default value.

NULL DELIM

Specifies that null values will not be enclosed by the delimiter character that is specified in DELIM *delimiter-character*.

To ensure that you can distinguish between empty values and null values, specify all three keywords.

The following table shows the results of different settings for a row that contains five columns that have the following values:

Col 1 BEGINNING (VARCHAR)

Col 2 NULL (Integer)

Col 3 NULL (Character 5)

Col 4 empty (VARCHAR)

Col 5 END (Character 3)

Table 28. Results of selected DELIMITED block keyword settings

DELIMITED block setting	Result
SEP ';'	BEGINNING;;;END
SEP ';' DELIM '/'	/BEGINNING/;/;/;/END/
SEP ';' DELIM '/' NULL DELIM	/BEGINNING/;/;/;/END/

The DATE, TIME, and TIMESTAMP columns are in ISO format and correspond to the following formats:

DATE YYYY-MM-DD

TIME HH.MM.SS

TIMESTAMP

YYYY-MM-DD-HH.MM.SS.NNNNNN

You can override the default DATE, TIME, and TIMESTAMP formats by specifying an OPTIONS block at the SELECT level. Only an OPTIONS block at the SELECT level is considered for this format.

VARIABLE block syntax and description:

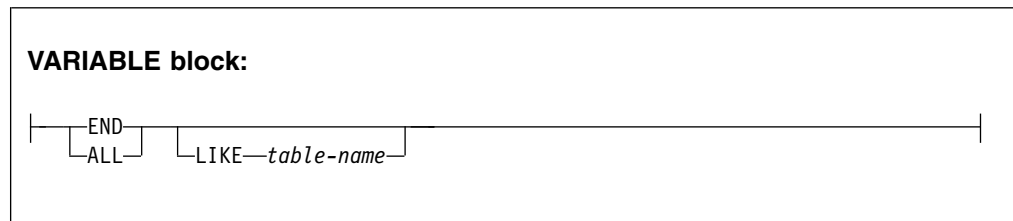
Use the VARIABLE block to indicate that the output data set must be compatible with the DB2 LOAD data set.

The default format of the output data set is variable block (VB), but you can specify fixed (F), fixed block (FB), or variable (V) in the JCL. DB2 HPU determines the LRECL at run time based on the following rules:

- If you specified RECFM=F or RECFM=FB, the LRECL must be larger than or equal to the sum of the lengths of the fields.
- If you specified RECFM=V, the LRECL must be larger than or equal to the sum of the lengths of the fields plus 4 bytes.

The length of variable-length fields is the sum of their maximum length plus 2 bytes.

The following syntax diagram shows the VARIABLE block:



END The characteristics and the sequence of fields in the output data set correspond to the characteristics and the sequence of fields in the SELECT statement.

The fields in the data set are also like the fields in the data set in DSNTIAUL format except for the following differences:

- The DATE, TIME, and TIMESTAMP columns are in ISO format and correspond to the following formats:

DATE YYYY-MM-DD

TIME HH.MM.SS

TIMESTAMP

YYYY-MM-DD-HH.MM.SS.NNNNNN

- If a column accepts nulls, the null indicator is generated at the beginning of the field. This indicator contains the value X'FF' if the field is null and X'00' if the value is usable.
- If the last-selected column is variable, the output data set is VB, and this column is written only on its effective length. The two length bytes are placed before the column.

You can override the default DATE, TIME, and TIMESTAMP formats by specifying an OPTIONS block at the SELECT level. Only an OPTIONS block at the SELECT level is considered for this format.

ALL All the variable columns are written using their actual length.

LIKE *table-name*

If the LIKE *table-name* clause is specified, DB2 HPU uses the characteristics of the table model as parameters and formats the data set to allow this table to be loaded.

The sequence of columns in the SELECT statement must match the columns in the model table. If format conversions are required, they follow DB2 HPU format rules.

The *table-name* must be in the format *user_ID*.TABLE. If the user ID is not specified, the name of the user (CURRENT SQLID) is used.

Field types are provided by the table model. A null-byte indicator is generated at the end of the field if the column of the table model can be null.

If a column in the SELECT statement is null and if the corresponding column in the table model does not allow nulls, the field in the output sequential data set has one of the following standard default values:

- 0 for numeric fields (INTEGER, SMALLINT, DECIMAL, FLOAT)
- A string of blank characters (CHAR, GRAPHIC)
- Two bytes of zero length (VARCHAR, LONG VARCHAR, VARGRAPHIC, LONG VARGRAPHIC)

- 0001-01-01 (DATE)
- 00.00.00 (TIME)
- 0001-01-01-00.00.00.000000 (TIMESTAMP)

You can override the default DATE, TIME, and TIMESTAMP formats by specifying an OPTIONS block at the SELECT level. Only an OPTIONS block at the SELECT level is considered for this format.

Related reference:

“DSNTIAUL block syntax and description” on page 193

Use the DSNTIAUL block to put the output data set in the same format that is produced by the DSNTIAUL program.

USER block syntax and description:

Use the USER block to indicate that the unloaded data is formatted according to the keywords that are specified in the USER block.

You can change field attributes for all selected columns, which means that you can specify several keywords for each column according to the type of data that the column contains.

The default values are determined by the values that are specified in the OPTIONS block.

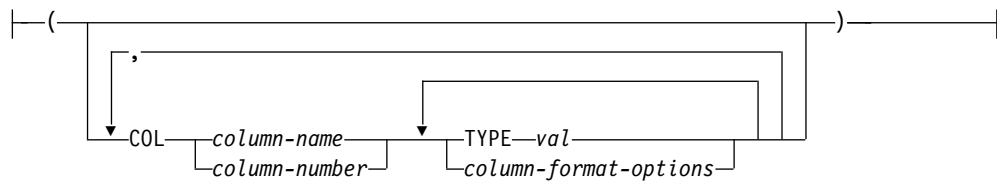
If all the unloaded fields are fixed, the default value of RECFM is FB. If at least one output field is variable, the default value of RECFM is VB.

If the LRECL is not specified, DB2 HPU determines it at run time based on the following rules:

- If you specified RECFM=F, the LRECL of the data set is equal to the sum of the maximum length of fields, regardless of the LRECL value in the JCL. The output data set is in FB format.
- If you specified RECFM=V or RECFM=VB, the LRECL of the data set is equal to the sum of the maximum length of fields plus 4 bytes, regardless of the LRECL value in the JCL. The output data set is in VB format.

The following syntax diagram shows the USER block:

USER block:



COL *column-name* | *column-number*

Specifies the name or number of the column in the SELECT statement.

You can specify the following attributes (in keywords) for each field:

A description of the keywords that specify these attributes follows:

TYPE *val*

Specifies the type of output field. The TYPE keyword specifies the

conversion to be performed. The following example shows how you might specify the type of output field: TYPE CHAR(10).

The default field format for output records is the format that was specified for columns in the SELECT statement.

column-format-options

Specifies the formatting options of the output data.

Related reference:

“Data types for output (TYPE keyword)” on page 458

The TYPE keyword of the SELECT statement (OPTION block for FORMAT USER) is used to create several types of data in the output. These types are declared in the keyword TYPE. The use of this keyword implies that data is to be converted from the original column type to the type that is declared in the TYPE keyword.

“Supported conversions” on page 462

DB2 HPU supports conversions to specific output data types.

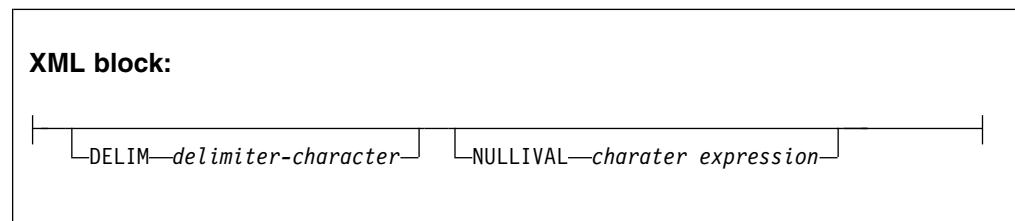
“OPTIONS block syntax and description” on page 134

Use the OPTIONS block to specify the default conversions that are with the SELECT statements. This block can be used in the GLOBAL block, the UNLOAD block, and the SELECT block.

XML block syntax and description:

Unloads data in XML format.

The following syntax diagram shows the format of the XML block:



DELIM *delimiter-character*

Specifies the delimiter character to be used to enclose CHAR, VARCHAR, GRAPHIC, and VARGRAPHIC fields in the output data set.

The value of the *delimiter-character* can be 'c' or X'hh'.

DELIM does not have a default value. The value is not delimited if no DELIM option is specified

NULL DELIM *character expression*

Specifies the delimiter character string to be unloaded when a NULL value is unloaded. This character string is not delimited even if the DELIM option is specified.

If NULLVAL not specified, the NULL value is not substituted and an empty string with no delimiter is unloaded.

SELECT statement examples

These SELECT statement examples show how to create sequential data sets in different formats, how to use the LISTDEF and TEMPLATE keywords, how to use the INTO clause and the REFORMAT clause, and how to use other DB2 HPU options.

The SELECT statement examples are based on the PERSONAL and PERSOBIS tables.

The following SQL statement shows how to create the PERSONAL table.

```
CREATE TABLE PERSONAL
( NAME CHARACTER(30) NOT NULL ,
  FNAME CHARACTER(15) ,
  AGE SMALLINT ,
  ADDRESS VARCHAR(100) NOT NULL WITH DEFAULT ,
  DATE_B DATE NOT NULL WITH DEFAULT ,
  SALARY DECIMAL(7,0) NOT NULL ,
  BONUS DECIMAL(7,0) NOT NULL WITH DEFAULT ) IN TABLESPACE DBNAME.TSNAME
```

The following SQL statement shows how to create the PERSOBIS table.

```
CREATE TABLE PERSOBIS
( WHOLE_NAME CHARACTER(40) NOT NULL ,
  AGE SMALLINT NOT NULL ,
  DATE_B DATE ,
  COMPENSATION INTEGER NOT NULL ) IN TABLESPACE DBNAME.TSNAME
```

Related reference:

“SELECT block syntax and description” on page 168

The SELECT statement specifies that a logical unload is to be done and indicates the parameters that are associated with the unload job. The SELECT block is a part of the UNLOAD block.

Example: Creating a sequential data set in USER format:

In this example, a sequential data set is created in USER format from the PERSONAL table.

```
UNLOAD TABLESPACE DBNAME.TSNAME DB2 NO
SELECT FNAME, NAME, ADDRESS, DATE_B , SALARY
FROM PERSONAL
OUTDDN (SYSUT1)
FORMAT USER (
  COL FNAME NULLID YES ,
  COL 3 TYPE CHARACTER(100) ,
  COL SALARY TYPE CHARACTER(7)
  JUST RIGHT ,
  COL 004 TYPE DATE_A )
```

In this example, data in the following three columns is changed:

- The ADDRESS column is converted to fixed format and its length is increased to 100 characters.
- The SALARY column is converted from decimal to display characters and it is aligned on the units position with the sign first.
- The DATE column is formatted to display in MM/DD/YY format.

A NULL byte is reserved before the NAME field.

The following output record format is produced:

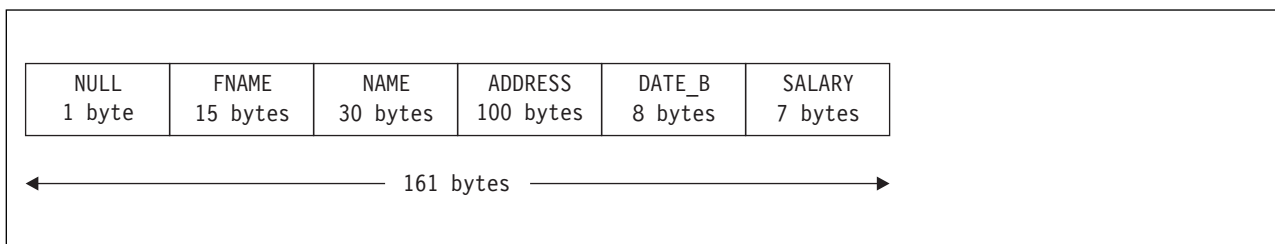


Figure 14. Output record format: sequential data set in USER format

By default, the output data set is in VARIABLE format with a maximum LRECL of 165, including the 4-byte field that describes the record of variable length. If the LRECL of the DD statement is greater than or equal to 161, you can produce a data set in fixed format.

Example: Creating a sequential data set in DSNTIAUL format:

In this example, a sequential data set is created in DSNTIAUL format from the columns of the PERSONAL table. The format of the PERSOBIS table is used.

```
UNLOAD TABLESPACE DBNAME.TSNAME DB2 NO
SELECT NAME , AGE , DATE_B , SALARY
FROM PERSONAL
OUTDDN SYSUT1
FORMAT DSNTIAUL
LIKE user.PERSOBIS
LOADDDN SYSUT2
```

In this example, data in the following columns is changed:

- The NAME column is changed from 30 to 40 characters. Blanks are added to the right.
- The DATE column is written in DB2 format.
- The SALARY column is converted to binary format.
- The AGE column in the SELECT statement can be null, but the AGE column of the PERSOBIS table cannot be null. If the column is null, the field is set to binary 0.

The data set with the SYSUT1 ddname contains the rows that were extracted. The data set with the SYSUT2 ddname contains the command for loading data by using the DB2 LOAD utility.

The following output record format is produced:

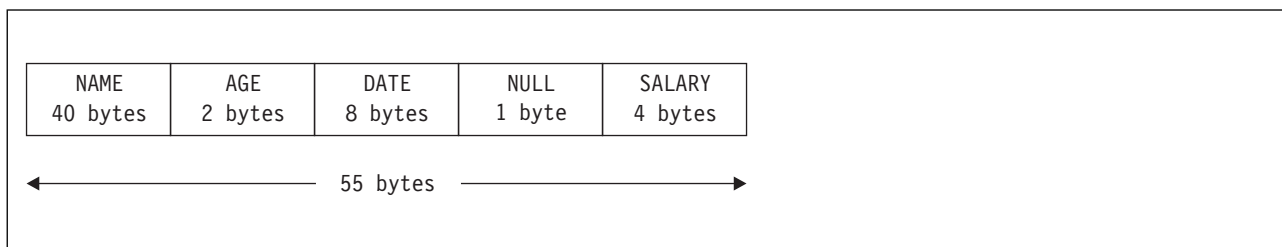


Figure 15. Output record format: sequential data set in DSNTIAUL format

The following data set loads SYSUT2 output, which was generated by DB2 HPU:

```
LOAD DATA INDDN SYSUT1
INTO TABLE PERSOBIS
(
  WHOLE_NAME          POSITION(1:40)
                      CHAR
,AGE                  POSITION(41:42)
                      SMALLINT
,DATE_B              POSITION(43:50)
                      DATE EXTERNAL      NULLIF((51)='?')
,COMPENSATION        POSITION(52:55)
                      INTEGER
)
```

Example: Creating a sequential data set that is compatible with the DB2 LOAD utility:

In this example, a sequential data set that is compatible with the DB2 LOAD utility is created.

```
UNLOAD TABLESPACE DBNAME.TSNAME DB2 NO
SELECT NAME , AGE , DATE_B , SALARY
  FROM PERSONAL
OUTDDN SYSUT1
FORMAT VARIABLE END
LIKE user.PERSOBIS
LOADDDN SYSUT2
```

The columns are extracted from the PERSONAL table by using the format of the PERSOBIS table.

In this example, data in the following columns is changed:

- The NAME column changes from 30 to 40 characters. Blanks are used to pad the column.
- The DATE column is written in ISO format.
- The SALARY column is converted to binary format.
- The AGE column in the SELECT statement can be null, but the AGE column of the PERSOBIS table cannot be null. If the column is null, the field is set to binary 0.

The SYSUT1 ddname contains rows that were extracted. The SYSUT2 ddname contains the command for loading data by using the DB2 LOAD utility.

The following output record format is produced:

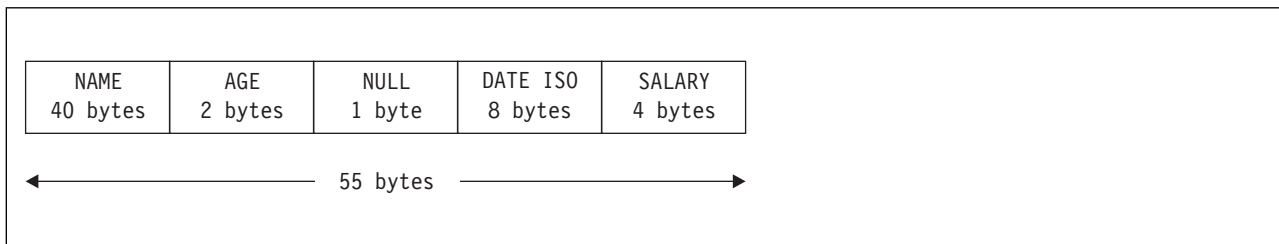


Figure 16. Output record format: sequential data set that is compatible with the DB2 LOAD utility

The following data set loads SYSUT2, which was generated by DB2 HPU:

```
LOAD DATA INDDN SYSUT1
INTO TABLE PERSOBIS
(
  WHOLE_NAME          POSITION(1:40)
                      CHAR
,AGE                  POSITION(41:42)
                      SMALLINT
,DATE_B              POSITION(44:51)
                      DATE EXTERNAL      NULLIF((43)=X'FF')
,SALARY              POSITION(52:55)
                      INTEGER
)
```

Example: Using TEMPLATESET when processing multiple UNLOAD and SELECT statements:

In this example, the TEMPLATESET option is used with user-defined variables in a TEMPLATE DSN expression.

The TEMPLATE OUT1 contains two user-defined variables, &CONST and &NUMBER. The &NUMBER variable is set in the GLOBAL OPTIONS block to the select number with a TEMPLATESET (NUMBER = :SELNUM). This setting is available for all UNLOAD commands and SELECT statements. The &CONST variable is set to a different value in each UNLOAD command by using a different TEMPLATESET option.

```
TEMPLATE OUT1 DSN HLQ.&CONST.&NUMBER. UNIT SYSDA
```

```
GLOBAL OPTIONS TEMPLATESET( NUMBER = :SELNUM )
```

```
UNLOAD TABLESPACE
```

```
OPTIONS TEMPLATESET( CONST = "C" )
```

```
SELECT * FROM TABLE1  
OUTDDN(OUT1)  
FORMAT DSNTIAUL
```

```
SELECT * FROM TABLE2  
OUTDDN(OUT1)  
FORMAT DSNTIAUL
```

```
UNLOAD TABLESPACE
```

```
OPTIONS TEMPLATESET( CONST = "D" )
```

```
SELECT * FROM TABLE3  
OUTDDN(OUT1)  
FORMAT DSNTIAUL
```

```
SELECT * FROM TABLE4  
OUTDDN(OUT1)  
FORMAT DSNTIAUL
```

Four files are dynamically allocated by using the OUT1 template. These files contain the following output data:

- The HLQ.C00001 file contains rows that were unloaded from TABLE1.
- The HLQ.C00002 file contains rows that were unloaded from TABLE2.
- The HLQ.D00001 file contains rows that were unloaded from TABLE3.
- The HLQ.D00002 file contains rows that were unloaded from TABLE4.

Example: Using LISTDEF and TEMPLATE statements:

In this example, LISTDEF and TEMPLATE statements are used to generate the logical unload of all the tables that are included in the table spaces that match the generic name DBTEST.*.

An output file that uses the OUTTEMP template and a LOAD SYSIN that uses the LOADTEMP template is generated for each table. A TEMPLATESET is used to set the value of the &SEL user-defined variable that is used in the template definitions to :SELNUM. This value is increased by one for each unloaded table starting with 00001.

```

LISTDEF LIST1 INCLUDE TABLESPACE DBTEST.*

TEMPLATE OUTTEMP DSN HLQ.&DB..&TS..T&SEL..UNLOAD
TEMPLATE LOADTEMP DSN HLQ.&DB..&TS..T&SEL..LOADD

GLOBAL OPTIONS TEMPLATESET( SEL = :SELNUM )

UNLOAD TABLESPACE
SELECT * FROM LIST(LIST1) OUTDDN OUTTEMP FORMAT DSNTIAUL LOADDN LOADTEMP

```

The following example uses LISTDEF and TEMPLATE statements on the DBTEST.TS1, DBTEST.TS2, DBTEST.TS3, and DBTEST.TS4 partitioned table spaces. TS1 and TS2 are unloaded using a separate file per partition. TS3 and TS4 are unloaded in a single output file. All partitions of TS1 and TS3 are unloaded, and only partitions 1 and 3 are unloaded for TS2 and TS4.

```

LISTDEF LIST1 INCLUDE TABLESPACE DBTEST.TS1
                INCLUDE TABLESPACE DBTEST.TS2 PARTLEVEL 1
                INCLUDE TABLESPACE DBTEST.TS2 PARTLEVEL 3

LISTDEF LIST2 INCLUDE TABLESPACE DBTEST.TS3
                INCLUDE TABLESPACE DBTEST.TS4 PARTLEVEL 1
                INCLUDE TABLESPACE DBTEST.TS4 PARTLEVEL 3

TEMPLATE OUTTEMP1 DSN HLQ.&DB..&TS..UNLOAD.P&PART.
TEMPLATE OUTTEMP2 DSN HLQ.&DB..&TS..UNLOAD
TEMPLATE LOADTEMP DSN HLQ.&DB..&TS..LOADD

UNLOAD TABLESPACE
SELECT * FROM LIST(LIST1) OUTDDN OUTTEMP1 FORMAT DSNTIAUL LOADDN LOADTEMP
SELECT * FROM LIST(LIST2) OUTDDN OUTTEMP2 FORMAT DSNTIAUL LOADDN LOADTEMP

```

The following files are generated:

Output files

- HLQ.DBTEST.TS1.UNLOAD.Pnnnnn (one file per partition)
- HLQ.DBTEST.TS2.UNLOAD.P00001
- HLQ.DBTEST.TS2.UNLOAD.P00003
- HLQ.DBTEST.TS3.UNLOAD
- HLQ.DBTEST.TS4.UNLOAD

LOADDDN files that contain the LOAD commands

- HLQ.DBTEST.TS1.LOADD
- HLQ.DBTEST.TS2.LOADD
- HLQ.DBTEST.TS3.LOADD
- HLQ.DBTEST.TS4.LOADD

Example: Using the REFORMAT clause to unload LOB columns with a LOB file reference:

In this example, the REFORMAT clause is used to unload data from tables that contain one or more LOB columns.

The tables are selected by using a LISTDEF statement. All tables from all table spaces that are contained in the DBTEST1 database are unloaded in FORMAT VARIABLE ALL.

The unloaded data is stored in a sequential file for standard columns and in a PDS by using a template and a LOB file reference for LOB columns. All output files, including the LOAD SYSIN, are allocated by using templates.

If some of the table spaces contain more than a single table, the &SEL user-defined variable is used in the DSN expression of the templates that are used in the OUTDDN and LOADDN files to distinguish the result of SELECT statements for tables that are located in the same table space.

For the OUTLOB template, the &TS variable is substituted with the name of the table space that stores the LOB column value, not the base table space name. This table space name is different for each LOB column of each table. Therefore, the DSN expression of template OUTLOB does not need to contain the &SEL variable.

```
LISTDEF LISTLOB INCLUDE TABLESPACE DBTEST1.*

TEMPLATE OUTDSN DSN HLQ.&DB..&TS..T&SEL..UNLOAD
TEMPLATE OUTLOB DSN HLQ.&DB..&TS..LOBDATA(&UNIQ.)
TEMPLATE LOADTEMP DSN HLQ.&DB..&TS..T&SEL..LOADD

GLOBAL OPTIONS REFORMAT ( TYPE CLOB INTO VARCHAR(54) CLOBF OUTLOB ,
                          TYPE BLOB INTO VARCHAR(54) BLOBF OUTLOB ,
                          TYPE DBCLOB INTO VARCHAR(54) DBCLOBF OUTLOB )

                          TEMPLATESSET( SEL = :SELNUM )

UNLOAD TABLESPACE
SELECT * FROM LIST(LISTLOB) OUTDDN OUTDSN FORMAT VARIABLE ALL LOADDN LOADTEMP
```

Example: Using the INTO clause in a SELECT statement to change the name and data type of some columns in the DSNTIAUL format:

In this example, the INTO clause of the SELECT statement is used to rename and reformat some of the columns from the DSN8810.EMP table.

The following column definitions are for the DSN8810.EMP table:

```
EMPNO      CHAR(6) NOT NULL,
FIRSTNME   VARCHAR(12) NOT NULL,
MIDINIT    CHAR(1) NOT NULL,
LASTNAME   VARCHAR(15) NOT NULL,
WORKDEPT   CHAR(3),
PHONENO    CHAR(4),
HIREDATE   DATE,
JOB        CHAR(8),
EDLEVEL    SMALLINT,
SEX        CHAR(1),
BIRTHDATE  DATE,
SALARY     DECIMAL(9, 2),
BONUS      DECIMAL(9, 2),
COMM       DECIMAL(9, 2)
```

When you specify the INTO clause, the SYSIN performs the following conversions:

- The PHONENO column is converted from CHAR(4) to SMALLINT.
- The HIREDATE column is converted from DATE to DATE EXTERNAL on 15 characters.
- The EDEVEL column is converted from SMALLINT to SMALLINT EXTERNAL on the default length of six characters.
- The SALARY, BONUS, and COMM columns are converted from DEC(9,2) to DECIMAL ZONED on nine, seven, and seven characters, respectively.

The LOAD SYSIN that is generated in the SYSPUNCH file contains a reload statement into the IBMUSER.EMP_BIS table with modified column names for the EMPNO and PHONENO columns.

```
UNLOAD TABLESPACE
DB2 NO QUIESCE YES QUIESCECAT NO
SELECT *
  INTO TABLE IBMUSER.EMP_BIS
      EMPLOYEE_NUMBER
      ,
      ,
      ,
      ,
      ,PHONE_NUMBER SMALLINT
      ,HIREDATE DATE EXTERNAL(15)
      ,
      ,EDLEVEL SMALLINT EXTERNAL
      ,
      ,
      ,SALARY DEC ZONED(9,2)
      ,BONUS DEC ZONED(7,2)
      ,COMM DEC ZONED(7,2)
FROM DSN8810.EMP
OUTDDN OUT
FORMAT DSNTIAUL LOADDN SYSPUNCH
```

The following example shows the SYSPUNCH file:

```
LOAD DATA
LOG NO REPLACE
EBCDIC CCSID(0037)

INTO TABLE IBMUSER.EMP_BIS
(
EMPLOYEE_NUMBER
  POSITION ( 1 )          CHAR ( 6 )
,
FIRSTNAME
  POSITION ( 7 )          VARCHAR
,
MIDINIT
  POSITION ( 21 )         CHAR ( 1 )
,
LASTNAME
  POSITION ( 22 )         VARCHAR
,
WORKDEPT
  POSITION ( 39 )         CHAR ( 3 )
  NULLIF( 42 ) = '?'
,
PHONE_NUMBER
  POSITION ( 43 )         SMALLINT
  NULLIF( 45 ) = '?'
,
HIREDATE_DEC
  POSITION ( 46 )         DATE EXTERNAL ( 15 )
  NULLIF( 61 ) = '?'
,
JOB
  POSITION ( 62 )         CHAR ( 8 )
  NULLIF( 70 ) = '?'
,
EDLEVEL
  POSITION ( 71 )         INTEGER EXTERNAL ( 6 )
  NULLIF( 77 ) = '?'
,
SEX
```

```

        POSITION (    78 )          CHAR (    1 )
        NULLIF(    79 ) = '?'
    ,
    BIRTHDATE
        POSITION (    80 )          DATE EXTERNAL (    10 )
        NULLIF(    90 ) = '?'
    ,
    SALARY
        POSITION (    91 : 99 )    DECIMAL ZONED
        NULLIF(   100 ) = '?'
    ,
    BONUS
        POSITION (   101 : 107 )    DECIMAL ZONED
        NULLIF(   108 ) = '?'
    ,
    COMM
        POSITION (   109 : 115 )    DECIMAL ZONED
        NULLIF(   116 ) = '?'
)

```

Example: Using an HFS file when unloading LOB data:

When you unload a table that contains LOB data, specify DSNTYPE(HFS) in the TEMPLATE statement, and specify a name expression that corresponds to an HFS file.

In this example, an HFS file for the LOB column is used when you unload the TBNAME01 table. The LOB file references are created in the existing /u/test/ directory with a file name that contains the record number, which is created because the :RECNUM keyword is specified.

```

TEMPLATE OUTTMPLT
DSN '/u/test/N&NUM.'
DSNTYPE HFS

UNLOAD TABLESPACE
DB2 NO
QUIESCE YES QUIESCECAT YES

OPTIONS
TEMPLATESET (NUM=:RECNUM)

SELECT COL_LOB
INTO COLOB VARCHAR(44) CLOBF OUTTMPLT
FROM userid.TBNAME01
OUTDDN (SYSREC01)

```

Example: Using temporal tables:

These examples show how you can unload reports with a period-specification clause in the SELECT statement or without the clause.

These examples unload reports about business activity that is based on business time columns with the following definitions. The definitions are excerpts from the CREATE TABLE statement.

- , C_BUS_START DATE NOT NULL
- , C_BUS_END DATE NOT NULL
- , PERIOD BUSINESS_TIME(C_BUS_START, C_BUS_END)

A SELECT statement with a period-specification clause

```

UNLOAD TABLESPACE
DB2 FORCE
SELECT C_INT_NN,
       'BUSINESS STARTED ON',
       C_BUS_START,
       'AND ENDED ON',
       C_BUS_END
FROM HPUDOC.TBDZAAPT201
FOR BUSINESS_TIME
AS OF '2010-04-30'
OUTDDN(DB2FRC)
FORMAT DELIMITED SEP ' '

```

A SELECT statement without a period-specification clause

```

UNLOAD TABLESPACE
DB2 NO
SELECT C_INT_NN,
       'BUSINESS STARTED ON',
       C_BUS_START,
       'AND ENDED ON',
       C_BUS_END
FROM HPUDOC.TBDZAAPT201
OUTDDN(DB2NO)
FORMAT DELIMITED SEP ' '      None

```

This report produces the following DB2NO file output data:

```

1 BUSINESS STARTED ON 2000-01-01 AND ENDED ON 2010-01-01
2 BUSINESS STARTED ON 2000-01-02 AND ENDED ON 2010-01-02

```

Example: Using timestamp precision:

These examples show how you can use timestamp precision.

In the following example, three columns with various timestamp precision are unloaded. C_TMSTP_n is defined as TIMESTAMP(n). The output format is TIMESTAMP_B.

```

UNLOAD TABLESPACE
DB2 NO
SELECT '
       C_TMSTP_0,
       C_TMSTP_6,
       C_TMSTP_12
FROM HPUDOC.TBDZAAPT98101
OUTDDN(TMSTMPB)
FORMAT DSNTIAUL
OPTIONS REFORMAT (TYPE TIMESTAMP INTO TIMESTAMP_B)

```

This example produces the following output data:

```

2005-06-01-07.31.06 2 005-06-01-07.31.06.123456 2005-06-01-07.31.06.123456789012

```

In the following example, three columns with various timestamp precision are unloaded. C_TMSTP_n defined as TIMESTAMP(n). Only the C_TMSTP_12 column is unloaded with the TIMESTAMP_B output format.

```

UNLOAD TABLESPACE
DB2 NO
SELECT
       C_TMSTP_0,
       C_TMSTP_6,
       C_TMSTP_12
FROM HPUDOC.TBDZAAPT98101

```



```

OUTDDN(OUTPUT)
FORMAT DSNTIAUL
OPTIONS REFORMAT (TYPE TIMESTAMP(12) INTO TIMESTAMP_B)

```

This example produces the following output data:

```
2005-06-01-07.31.06 2005-06-01-07.31.06.123456 20050601073106123456789012
```

Example: Maximizing resources during sort operations:

In this example, part of the SYSIBM.SYSCOPY table is unloaded, and the output data is sorted. Because the WHERE clause filter ratio is low, MAX_EXPECTED_ROWS is specified so that only the necessary resources (work space) are used to process the sort operation.

```

UNLOAD TABLESPACE
SELECT ICDATE
FROM SYSIBM.SYSCOPY
WHERE ICDATE > '110420'
ORDER BY DBNAME, TSNAME
OUTDDN(OUTPUT)
MAX_EXPECTED_ROWS 2000
FORMAT DSNTIAUL
LOADDDN SYSPUNCH

```

Example: The SYSPRINT data set

The SYSPRINT data set contains detailed information about the unloaded objects.

The following examples show a SYSPRINT data set that corresponds to a table space with one table, a SYSPRINT data set that corresponds to a partitioned table space with one table, and a report that is generated when output files are written on tapes.

A SYSPRINT data set that corresponds to a table space with one table

The following example shows a SYSPRINT data set that corresponds to the unloaded DBINFDM.TSSCA table space, which contains a single table that is named *user.TBSCA*. This table contains 5002578 rows. If the return code is 0, no action is required. If the return code is not 0, look in the SYSPRINT for warning and error messages.

```

INZU224I IBM DB2 HIGH PERFORMANCE UNLOAD V4.1
INZI175I PROCESSING SYSIN AS EBCDIC.
-----1-----2-----3-----4-----5-----6-----7-----8
000001 UNLOAD TABLESPACE DBINFDM.TSSCA
000002 DB2 NO LOCK NO QUIESCE YES QUIESCECAT NO
000003 SELECT * FROM TBSCA
000004 OUTDDN (SYSREC00 )
000005 FORMAT DSNTIAUL
000006 LOADDDN SYSPUNCH

DSNU000I  DSNUGUTC - OUTPUT START FOR UTILITY, UTILID = DB2UNLOAD
DSNU050I  DSNUGUTC - QUIESCE TABLESPACE DBINFDM.TSSCA
DSNU477I  -DZ7M DSNUQUA - QUIESCE SUCCESSFUL FOR TABLESPACE DBINFDM.TSSCA
DSNU474I  -DZ7M DSNUQUA - QUIESCE AT RBA 0014C98DF148 AND AT LRSN 0014C98DF148
DSNU475I  DSNUQUIB - QUIESCE UTILITY COMPLETE, ELAPSED TIME= 00:00:00
DSNU010I  DSNUGBAC - UTILITY EXECUTION COMPLETE, HIGHEST RETURN CODE=0
INZU180I  UTPRINT DD CARD IN JCL IS NOT USED WHEN VUX020/SORTCLAS IS SPECIFIED IN THE
          PARM LIB OR WHEN SORTCLASS IS SPECIFIED IN SYSIN.

-----
INZU281I - UNLOAD STARTING AT POS(1, 1)

INZU277I - PROCESSING UNLOAD 00001 FROM TABLESPACE DBINFDM.TSSCA
INZU279I - SELECT STATEMENTS USING SINGLE TABLE SPECIFICATION

```

```

INZU280I - SELECT 00001 STARTING AT POS(3, 1)
INZU282I  OUTPUT DDNAME=SYSREC00
INZX090 SYSREC00 : 5002578 RECORDS WRITTEN

```

```

- TABLESPACE DBINFDM.TSSCA - DB2 HIGH PERFORMANCE UNLOAD - STATISTICS - 10/20/06
* CREATOR.TABLE * OBID * ROWS READ * ROWS KEPT * TS PG READ *
*-----*-----*-----*-----*-----*-----*
* IBMUSER.TBSCA * 345 * 5002578 * 5002578 * *
*-----*-----*-----*-----*-----*-----*
* TOTAL UNLOAD STATISTICS ...* * 5002578 * 5002578 * 201103 *
* INVALID ROWS.....* * 0
* NUMBER OF PAGES IN ERROR...* * 0

```

```

INZU222I SYSREC00, TOTAL NUMBER OF RECORDS WRITTEN 5002578
INZU260I GENERATING LOAD STATEMENT FOR SELECT STARTING AT POS(3, 1)

```

A SYSPRINT data set that corresponds to a partitioned table space with one table

The following example shows a SYSPRINT data set that corresponds to the DBDM7M.TSTEST16 partitioned table space, which has six partitions and contains a single table that is named IBMUSER.TSTEST16. This table contains 1500000 rows. The ORDER CLUSTER clause unloads the data in the order of the clustering index. A template allocates a separate file per partition.

```

INZI175I PROCESSING SYSIN AS EBCDIC.
-----1-----2-----3-----4-----5-----6-----7-----8
000001 TEMPLATE DDNOUT DSN IBMUSER.&DB..&TS..&PART..&TIME. UNIT WORK
000002 UNLOAD TABLESPACE
000003 DB2 NO LOCK NO QUIESCE NO QUIESCECAT NO
000004 SELECT * FROM TSTEST16 ORDER CLUSTER
000005 OUTDDN (DDNOUT )
000006 FORMAT DSNTIAUL
000007 LOADDN SYSPUNCH

```

```

INZU180I UTPRINT DD CARD IN JCL IS NOT USED WHEN VUX020/SORTCLAS IS SPECIFIED IN THE
PARMLIB OR WHEN SORTCLASS IS SPECIFIED IN SYSIN.

```

```

-----
INZU281I - UNLOAD STARTING AT POS(2, 1)

```

```

INZU277I - PROCESSING UNLOAD 00001 FROM TABLESPACE DBDM7M.TSTEST16
INZU279I - SELECT STATEMENTS USING SINGLE TABLE SPECIFICATION
INZU280I - SELECT 00001 STARTING AT POS(4, 1)
INZI329I - DATASET ALLOCATED FOR PARTITION 00001. TEMPLATE= DDNOUT
DDNAME=SYS00000
DSN=IBMUSER.DBDM7M.TSTEST16.P00001.T170258
INZI329I - DATASET ALLOCATED FOR PARTITION 00002. TEMPLATE= DDNOUT
DDNAME=SYS00001
DSN=IBMUSER.DBDM7M.TSTEST16.P00002.T170258
INZI329I - DATASET ALLOCATED FOR PARTITION 00003. TEMPLATE= DDNOUT
DDNAME=SYS00002
DSN=IBMUSER.DBDM7M.TSTEST16.P00003.T170258
INZI329I - DATASET ALLOCATED FOR PARTITION 00004. TEMPLATE= DDNOUT
DDNAME=SYS00003
DSN=IBMUSER.DBDM7M.TSTEST16.P00004.T170258
INZI329I - DATASET ALLOCATED FOR PARTITION 00005. TEMPLATE= DDNOUT
DDNAME=SYS00004
DSN=IBMUSER.DBDM7M.TSTEST16.P00005.T170258
INZI329I - DATASET ALLOCATED FOR PARTITION 00006. TEMPLATE= DDNOUT
DDNAME=SYS00005
DSN=IBMUSER.DBDM7M.TSTEST16.P00006.T170258
INZU221I PARALLELISM ACTIVATED FOR PARTITION PROCESSING.
MAXIMUM DEGREE OF PARALLELISM IS SET TO 9
INZX090 SYS00003 : 250020 RECORDS WRITTEN
INZX090 SYS00002 : 249990 RECORDS WRITTEN
INZX090 SYS00004 : 249990 RECORDS WRITTEN
INZX090 SYS00000 : 249990 RECORDS WRITTEN
INZX090 SYS00005 : 250020 RECORDS WRITTEN
INZX090 SYS00001 : 249990 RECORDS WRITTEN

```

```

- TABLESPACE DBDM7M.TSTEST16 - DB2 HIGH PERFORMANCE UNLOAD - STATISTICS - 10/20/06
* CREATOR.TABLE * PART/IXSC * ROWS READ * ROWS KEPT * TS PG READ * IX ROWS READ *
*-----*-----*-----*-----*-----*-----*
* IBMUSER.TSTEST16 * 1/100% * 249990 * 249990 * 7578 * 249990 *
* OBID=302 * 2/100% * 249990 * 249990 * 7578 * 249990 *
* * 3/100% * 249990 * 249990 * 7578 * 249990 *
* * 4/100% * 250020 * 250020 * 7579 * 250020 *
* * 5/100% * 249990 * 249990 * 7578 * 249990 *
* * 6/100% * 250020 * 250020 * 7579 * 250020 *
*-----*-----*-----*-----*-----*-----*
* TOTAL UNLOAD STATISTICS ...* 6/100% * 1500000 * 1500000 * 45470 * 1500000 *
* INVALID ROWS.....* * 0
* NUMBER OF PAGES IN ERROR...* * 0

```

```
INZU222I SYS00000, TOTAL NUMBER OF RECORDS WRITTEN 249990
INZU222I SYS00001, TOTAL NUMBER OF RECORDS WRITTEN 249990
INZU222I SYS00002, TOTAL NUMBER OF RECORDS WRITTEN 249990
INZU222I SYS00003, TOTAL NUMBER OF RECORDS WRITTEN 250020
INZU222I SYS00004, TOTAL NUMBER OF RECORDS WRITTEN 249990
INZU222I SYS00005, TOTAL NUMBER OF RECORDS WRITTEN 250020
INZU260I GENERATING LOAD STATEMENT FOR SELECT STARTING AT POS(4, 1)
```

The following information is shown in the data set:

CREATOR.TABLE

Indicates the name of the table that has been unloaded. For partitioned table spaces, the OBID is displayed under the table name.

OBID/PART NO./IXSC

OBID Indicates the OBID value for each table in a non-partitioned table space.

PART NO./PART

Indicates the partition number for a partitioned table space. The OBID of the unloaded table is indicated under the table name in the CREATOR.NAME column. The value on the TOTAL UNLOAD STATISTICS line contains the total number of partitions in the table space. The number of partitions in the table space can differ from the number of partitions that are unloaded when PART is used in the UNLOAD command or PARTLEVEL is used in a LISDEF.

IXSC Indicates the percentage of rows that were located in the correct order based on the clustering index when an index scan is used. This column is displayed only if an ORDER BY corresponds to the order of the clustering index or if an ORDER CLUSTER clause has been coded.

ROWS READ

Indicates the total number of rows that were read. The number on the TOTAL UNLOAD STATISTICS line can be greater than the number of rows that were read for a single table. For example, for a multi-table table space, the total number of rows that were read also includes the rows of the tables that were dropped. For a partitioned table space, each line corresponds to the number of rows that were read for the partition, and the total corresponds to the total number of rows that were read for the table space.

ROW KEPT

Indicates the total number of rows that were kept after validation. This number is identical to the ROWS READ value.

TS PG READ

Indicates the number of pages that were read for each partition and the total for all the selected partitions on the TOTAL UNLOAD STATISTICS line for partitioned table spaces. For nonpartitioned table spaces, only the value on the TOTAL UNLOAD STATISTICS line is displayed for the complete table space.

IX ROWS READ

Indicates the number of records that were read from the clustering index for each partition and the total number of records on the TOTAL UNLOAD STATISTICS line. This column is displayed only when the index scan is used.

TOTAL UNLOAD STATISTICS

Shows total number of rows that were processed.

Report when output files are written on tapes

The following example shows a generated report when output files are written on tapes. Information about files that span across several volumes is documented only in the section that is related to the last volume that the files have been written on. This type of report is displayed only if TAPEREPOR YES is specified, or if the value of the VZM11/TAPERPT parameter is YES.

```
- TAPE SUMMARY - STATISTICS - 08/20/10
* VOLUME = TAP040 *
*-----*
* DSN = MZLFRD.AEST.DBTLS00.RP1XS9FN * FILESEQ = 00000001 *
* DSN = MZLFRD.AEST.DBTLS00.RP1XS9G4 * FILESEQ = 00000002 *
* DSN = MZLFRD.AEST.DBTLS00.RP1XS9HB * FILESEQ = 00000003 *
* DSN = MZLFRD.AEST.DBTLS00.RP1XS9IA * FILESEQ = 00000004 *
* DSN = MZLFRD.AEST.DBTLS00.RP1XS9J0 * FILESEQ = 00000005 *
* DSN = MZLFRD.AEST.DBTLS00.RP1XS9KA * FILESEQ = 00000006 *
* DSN = MZLFRD.AEST.DBTLS00.RP1XS9KL * FILESEQ = 00000007 *
* DSN = MZLFRD.AEST.DBTLS00.RP1XS9LC * FILESEQ = 00000008 *
* DSN = MZLFRD.AEST.DBTLS00.RP1XS9LM * FILESEQ = 00000009 *
* . . . *
*-----*
* VOLUME = TAP030 *
*-----*
* DSN = MZLFRD.AEST.DBTLS00.RP1XTCCI * FILESEQ = 00000177 *
* DSN = MZLFRD.AEST.DBTLS00.RP1XTCDQ * FILESEQ = 00000178 *
* DSN = MZLFRD.AEST.DBTLS00.RP1XTCD1 * FILESEQ = 00000179 *
* DSN = MZLFRD.AEST.DBTLS00.RP1XTCEH * FILESEQ = 00000180 *
```

Chapter 5. DB2 HPU interactive component

The DB2 HPU interactive component can be used to generate the DB2 HPU batch utility job. Although using the interactive application is not required, it can be useful for inexperienced users in generating the batch utility job.

You can use either the DB2 Administration Tool or the DB2 Tools Launchpad to start the DB2 HPU interactive component.

DB2 HPU interactive component primary commands

Primary commands are used with the DB2 HPU interactive component.

Command	Description
COLUMNS	Displays the SELECT Columns panel. For more information, see “Selecting columns for the SELECT statement” on page 238.
COPYDDN	Displays the COPYDDN File Parameters panel. This panel is used to specify parameters for the image copy file. This command is available only if a table space was selected. For more information, see “Specifying COPYDDN file parameters” on page 222.
FORMAT	Displays the SELECT Format panel, which is used to specify the output format for the SELECT statement. For more information, see page “Specifying the format of the SELECT statement” on page 234.
FUSER	Displays the Select User Format panel. For more information, see “Displaying and modifying the USER format of the SELECT statement” on page 241. The FUSER command (Select user format) command is available only if you select option 4 in the FORMAT field on the SELECT Format panel, as shown in Figure 36 on page 234.
GENERAL	Displays the General Options panel. This panel is used to specify general options for a DB2 HPU member. For more information, see “Specifying general options” on page 218 for more information.
JCL	Displays the JCL panel, which is used to generate JCL. This JCL can either be modified under the editor screen, saved, or directly submitted. The JCL command is available only if a table space was selected. For more information, see “Generating JCL to run a DB2 HPU job” on page 249.

Command	Description
LOADDDN	Displays the LOADDDN File panel. For more information, see “Displaying and modifying LOADDDN file parameters” on page 248.
OPTIONS	Displays the SELECT Statement Option panels, which are used to specify values for columns of type DATE, TIME, or TIMESTAMP. For more information, see “Specifying options for the SELECT statement” on page 225.
ORDERBY	Displays the SELECT Order By panel. For more information, see “Sorting columns of a SELECT statement” on page 240.
OUTDDN	Displays the OUTDDN File panel. For more information, see “Displaying the list of OUTDDN files” on page 245.
PART	Displays the Partition List panel, which is used to select partitions that are to be unloaded. This command is available only if a partitioned table space was selected. For more information, see “Including and excluding partitions” on page 221.
SELECT	Displays the SELECT Statement List panel. For more information, see “Updating the list of SELECT statements” on page 228.
SELECTALL	Selects all columns.
UNLDDN	Displays the UNLDDN File Parameters panel, which is used to specify parameters for the allocation of the data set that contains the data of unloaded table spaces or partitions. For more information, see “Specifying UNLOAD file parameters” on page 224.
WHERE	Displays the ISPF edit panel so you can enter an SQL statement.

Related tasks:

“Specifying the format of the SELECT statement” on page 234

The SELECT Format panels are used to specify the output format for a SELECT statement.

“Selecting columns for the SELECT statement” on page 238

The SELECT Columns panel is used to select columns for the SELECT statement.

“Sorting columns of a SELECT statement” on page 240

The SELECT ORDER BY panel is used to specify the order in which each column in the SELECT statement is processed for the output data set and whether to sort the column in ascending or descending order.

“Displaying and modifying the USER format of the SELECT statement” on page 241

The Select user format panels are used to update the USER formats.

“Displaying the list of OUTDDN files” on page 245

The OUTDDN list panel is used to display the list of OUTDDN files.

“Displaying and modifying LOADDDN file parameters” on page 248
The LOADDDN file panel is used to display and modify LOADDDN file parameters.

“Generating JCL to run a DB2 HPU job” on page 249

You can generate JCL, edit JCL, submit JCL, save JCL, and enter job card parameters for a DB2 HPU job by invoking the JCL command.

Starting the DB2 HPU interactive component in stand-alone mode

You can start DB2 HPU interactive component in stand-alone mode without using the DB2 Administration Tool or the DB2 Tools Launchpad.

Procedure

1. Use one of the following methods:

- Run the INZHPU procedure that is located in the SINZCLST library. When you run the INZHPU procedure in stand-alone mode, do not pass an argument.

- Issue this command:

```
EXEC 'HINZnnn.SINZCLST(INZHPU)'
```

In this example, HINZnnn.SINZCLST is the name of the library that contains the INZHPU procedure. You can also associate this command with an option in a menu.

Restriction: The TSO user ID that runs this command requires a minimum REGION of 4096K.

- Issue this command:

```
TSO INZHPU
```

Requirement: To use this method, you must have installed INZHPU as a TSO command.

2.

Related tasks:

“Optional: Creating a TSO command for the DB2 HPU interactive component” on page 14

You can make the DB2 HPU interactive component available as a TSO command.

Starting the DB2 HPU interactive component by using the DB2 Administration Tool

The DB2 HPU interactive component can be started by using the DB2 Administration Tool.

Procedure

Specify the **HPU** line command on the DB2 Admin panel. DB2 HPU interactive component can be started from either the Table Spaces panel or from the Tables, Views, and Aliases panel.

If you invoke the **HPU** command from the Tables, Views, and Aliases panel, the SELECT Format panel is displayed. For more information about the SELECT Format panel, see “Specifying the format of the SELECT statement” on page 234.

```

DB2 Admin ----- DSN1 Table Spaces ----- Row 1 of 269
Command ==> Scroll ==> CSR

Valid line commands are:
T - Tables D - Database A - Auth G - Storage group ICS - Image copy status
DIS - Display database STA - Start database STO - Stop database X - Indexes
? - Show all line commands

```

Select	Name	Owner	DB Name	BP	L	E	S	I	C	Ntable	N	Active	Segsz	T	L
*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
	SYSCOPY	SYSIBM	DSNDB06	BP0	A	N	A	N	N	1		0	0	Y	
	SYSDBASE	SYSIBM	DSNDB06	BP0	A	N	A	N	N	14		0	0	Y	
HPU	SYSDBAUT	SYSIBM	DSNDB06	BP0	A	N	A	N	N	2		0	0	Y	
	SYSDDF	SYSIBM	DSNDB06	BP0	P	N	A	N	N	7		0	4	Y	
	SYSGPAUT	SYSIBM	DSNDB06	BP0	A	N	A	N	N	1		0	0	Y	
	SYSGROUP	SYSIBM	DSNDB06	BP0	A	N	A	N	N	2		0	0	Y	
	SYSOBJ	SYSIBM	DSNDB06	BP0	P	N	A	N	N	8		0	4	Y	
	SYSPKAGE	SYSIBM	DSNDB06	BP0	A	N	A	N	N	8		0	4	Y	
	SYSPLAN	SYSIBM	DSNDB06	BP0	A	N	A	N	N	5		0	0	Y	
	SYSSTATS	SYSIBM	DSNDB06	BP0	A	N	A	N	N	6		0	4	Y	
	SYSSTR	SYSIBM	DSNDB06	BP0	A	N	A	N	N	4		0	4	Y	
	SYSUSER	SYSIBM	DSNDB06	BP0	A	N	A	N	N	1		0	0	Y	
	SYSVIEWS	SYSIBM	DSNDB06	BP0	A	N	A	N	N	4		0	0	Y	

Figure 17. DB2 Administration Table Spaces panel

If you invoke the **HPU** command from the Table Spaces panel, the General Options panel is displayed. See “Specifying general options” on page 218 for more information about the General Options panel.

```

DB2 Admin ----- DSN1 Tables, Views, and Aliases ----- Row 1 of 705
Command ==> Scroll ==> CSR

Valid line commands are:
C - Columns A - Auth L - List X - Indexes S - Table space D - Database
V - Views T - Tables P - Plans Y - Synonyms SEL - Select prototyping
? - Show all line commands

```

Sel	Name	Owner	T	DB Name	TS Name	Cols	Rows	Checks
*	*	*	*	*	*	*	*	*
	SYSCOPY	SYSIBM	T	DSNDB06	SYSCOPY	22	-1	0
	SYSFIELDS	SYSIBM	T	DSNDB06	SYSDBASE	13	-1	0
	SYSTABLESPACE	SYSIBM	T	DSNDB06	SYSDBASE	37	-1	2
HPU	SYSTABLES	SYSIBM	T	DSNDB06	SYSDBASE	43	-1	4
	SYSTABLEPART	SYSIBM	T	DSNDB06	SYSDBASE	33	-1	0
	SYSTABAUTH	SYSIBM	T	DSNDB06	SYSDBASE	29	-1	0
	SYSYNONYMS	SYSIBM	T	DSNDB06	SYSDBASE	7	-1	0
	SYSRELS	SYSIBM	T	DSNDB06	SYSDBASE	13	-1	0
	SYSLINKS	SYSIBM	T	DSNDB06	SYSDBASE	12	-1	0
	SYSKEYS	SYSIBM	T	DSNDB06	SYSDBASE	7	-1	0
	SYSINDEXPART	SYSIBM	T	DSNDB06	SYSDBASE	26	-1	0
	SYSINDEXES	SYSIBM	T	DSNDB06	SYSDBASE	38	-1	7
	SYSFOREIGNKEYS	SYSIBM	T	DSNDB06	SYSDBASE	7	-1	ã

Figure 18. DB2 Admin Tables, Views and Aliases panel

Related tasks:

“Specifying the format of the SELECT statement” on page 234
The SELECT Format panels are used to specify the output format for a SELECT statement.

“Specifying general options” on page 218
You can use the General Options panel to specify DB2 HPU general options.

Starting DB2 HPU interactive component by using the DB2 Tools Launchpad

The DB2 HPU interactive component can be started by using the DB2 Tools Launchpad.

Procedure

1. Specify the **S** line command on the DB2 Tools Launchpad panel, as shown in the following example. Press Enter.

The DB2 System panel is displayed, as shown in the following figure:

```
----- DB2 Tools Launchpad ----- Row 1 of 8
Command ==>                               Scroll ==> CSR
Select the DB2 tool you wish to launch or enter its code in the command line.
Sel Code  Tool Name                               Rel  Prog No.
---      -
ADM      DB2 Administration Tool                 710  5697-L90
---      -
s  HPU    High Performance Unload                 410  5655-AA1
---      -
---      - PERFORMANCE MANAGEMENT TOOLS -----
        No table entries in this category
---      - RECOVERY AND REPLICATION MANAGEMENT TOOLS --
        No table entries in this category
***** Bottom of data *****
```

Figure 19. DB2 Tools Launchpad panel

```
INZ0005 ----- DB2 System ----- 11:44
Command ==>

Enter :
*DB2 system name . . ____ (mandatory field)

                                     ( PF3 to exit )
```

Figure 20. DB2 subsystem name panel

2. Specify a DB2 subsystem name, and press Enter. The Database And Tablespace Selection panel is displayed, as shown in the following figure:

```

INZ0010 ----- DZ7M - Database And Table Space Selection ----- Row 1 of 15
Command ==> Scroll ==> CSR

Commands: CANCEL  SELECT

Line commands: S - Select a table space

Generic:
Database . . . DBDM7M%          Table space . . . @(#) 02 1.39@(#)

                                                                    ( PF3 to exit )

Select  DB Name  TS Name
-----  -
.       DBDM7M  DSNRSTAT
.       DBDM7M  SEAVDT
.       DBDM7M  TBMASKCH
.       DBDM7M  TSFLOAT
.       DBDM7M  TSJCD16A
.       DBDM7M  TSMASK
.       DBDM7M  TSPART1
.       DBDM7M  TSPAR10
.       DBDM7M  TSPAR100
.       DBDM7M  TSPAR250
.       DBDM7M  TSPAR254
.       DBDM7M  TSSCA
.       DBDM7M  TSSCA2
.       DBDM7M  TS3401A
.       DBDM7M  TS3474A

***** Bottom of data *****

```

Figure 21. Database And Tablespace Selection panel

3. Use one of the following methods to select table spaces:
 - Issue the SELECT primary command to select one or more tables.
 - Issue the Select line command to select one table space. When you use the Select line command, the General options panel is displayed, which is used to specify parameters for the table space that you selected.
 - Specify the name of a database in the **Database ==>** field and the name a table space in the **Tablespace ==>** field.

Results

If you specify non-generic criteria and you do not include a percent sign (%) when table spaces are listed by generic names, a percent sign is automatically added to the end of the field.

Related tasks:

“Selecting tables” on page 231

The Table List panel is used to create SELECT statements.

Specifying general options

You can use the General Options panel to specify DB2 HPU general options.

Procedure

1. Specify the **HPU** command on the DB2 Admin panel or after you specify a DB2 subsystem and table space. The General Options panel is displayed, as shown in the following figure:

```

INZA10      ----- General Options ----- 11:46
Command ==>

Commands : PART   COPYDDN   OPTIONS   SELECT   UNLDDN   JCL

DB2 system name . : DZ7M      Object name . : DBDM7M   . TSEXT

Utility ID . . . HPU
Part . . . . . * _____ (L - list, * - all, nnn - partition number)
                0 partition(s) selected from 0 total partitions

DB2 . . . . . _ (Y - Yes, N - No, F - Force)
LOCK . . . . . _ (Y - Yes, N - No)
QUIESCE . . . . _ (Y - Yes, N - No)
QUIESCECAT . . . _ (Y - Yes, N - No)
UNLMAXROWS . . . _____
UNLREQROWS . . . _____

                                     ( PF3 to exit)

```

Figure 22. General Options panel

2. Specify values for each of the following fields:

Utility ID

Specify the prefix that is used to build the unique Utility Identifier for the DB2 HPU job. The maximum length of the prefix is 11 bytes. This parameter is required.

Part Specify information about the partitions of a table space.

- L** Indicates that several partitions were selected by using the **PART** command. The **L** option is not used to display the list of partitions.
- *** Selects all partitions of a table space.
- nnn** Indicates the number of a single partition of the table space.

DB2 Specify whether a given request can or must be processed through DB2.

No The **SELECT** statements must be processed directly by DB2 HPU. If a **SELECT** statement is not supported by HPU, an error is generated and the program stops.

Attention: The entire **SYSIN** file is parsed before any unload command is processed.

Yes If the **SELECT** statement is too complex to be handled directly by the product, DB2 is called to extract the rows.

Force DB2 must be used to extract the requested rows. This process is useful when the **SELECT** statement uses a filtering predicate that is efficiently processed through SQL and when the filtering factor is high.

Restriction: DB2 **FORCE** cannot be used when attempting to unload from an image copy. Attempting to use DB2 **FORCE** to unload from an image copy results in an error and the program stops.

LOCK Specify the lock mode of the table space.

- Yes** The table space is accessed in read only mode when the DB2 HPU job is running.
- No** The table space is processed without changing its access mode.

QUIESCE

Specify the processing of the QUIESCE command.

- Yes** A QUIESCE command is processed if the table space is not in the COPY pending status; otherwise, processing of the QUIESCE command is stopped and restarted.
- No** The table space is processed without a QUIESCE command.

Attention: DB2 HPU operates on the physical VSAM data set level that is outside of DB2. If you run DB2 HPU on a table where a row was just inserted, it is possible that the unloaded data might not contain the row that was inserted. The unloaded data does not show the row because DB2 might not have externalized the data to DASD yet. This situation can happen when you use DB2 HPU without issuing a QUIESCE (or STOP) on the object. Be careful when using QUIESCE NO.

QUIESCECAT

This parameter is used only if the QUIESCAT parameter in the PARMLIB was set to YES or NO to specify that a QUIESCE command is to be processed on the catalog tables. The following values are valid:

- Yes** A quiesce point is taken for the table spaces that are listed for the parameter VUM014 before running DB2 HPU.
- No** No quiesce point is taken on these table spaces. This keyword is not required, and the QUIESCAT PARMLIB value becomes the default value.

UNLMAXROWS

Specify the maximum number of rows to be extracted for the physical unload. If the process involves a partitioned table space treated partition-by-partition, the limit applies to each partition.

UNLFREQROWS

Specify the sampling frequency for the physical unload. One row out of every *n* rows is. The value *n* is the argument of UNLFREQROWS.

- PF3** Press PF3 to end the current unload JCL setup process. The following figure shows the Confirm to leave the current Unload panel. Specify Y to cancel the unload JCL setup process or N to continue.

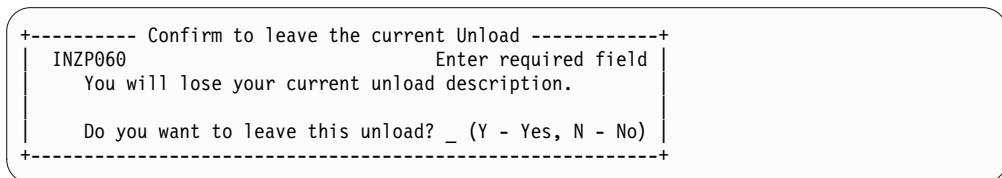


Figure 23. Confirm to leave the current Unload panel

What to do next

Specify a primary command in the command field to issue the request and generate the output, or press Enter to access the next panel.

Related reference:

“DB2 HPU interactive component primary commands” on page 213
 Primary commands are used with the DB2 HPU interactive component.

Including and excluding partitions

The Partition List panel is used to include or exclude one or more partitions for a DB2 HPU job.

Procedure

1. Specify the **PART** command on the General Options panel and press Enter. The Partition List panel is displayed, as shown in the following figure:

```

INZA201 ----- DZ7M DBDM7M . TSTEST16 - Partition List ----- Row 1 of 7
Command ==>                                         Scroll ==> CSR

Commands : GENERAL   COPYDDN   OPTIONS   SELECT   UNLDDN   JCL

Line commands: S - Select partition, U - Unselect, Z - Zoom

                                                                    ( PF3 to exit)

S Part  Selected High value of partition
-----
_ ALL   YES
_ 0001   8333
_ 0002   16666
_ 0003   24999
_ 0004   33333
_ 0005   41666
_ 0006   99999
***** Bottom of data *****
  
```

Figure 24. Partition List panel

The Partition List panel contains the following columns.

S Use this column to specify the partitions to be selected or deselected. Enter a selection code in this column.

Part This column contains the available partitions.

To select a partition, specify the S selection code beside a partition number.

To deselect a partition, specify the U selection code beside a partition number.

To select all partitions, specify the S selection code beside **ALL** in the Part column.

Requirement: You must select at least one partition.

Selected

YES indicates that the partition was selected.

High value of partition

Displays the first 57 characters of the LIMITKEY column of the SYSIBM.SYSTABLEPART table.

2. Specify the appropriate line command. You can specify the following line commands on the Partition List panel:

S Select one or more partitions.

- U** Undo the selection of one or more partitions.
- Z** Displays the Zoom Partition panel, as shown in the following figure. Select this option when the value of the High value of partition column is too high to display in the field. The Zoom Partition panel shows the partition number, indicates if it was selected, and displays the entire value for the partition.

```

INZA21 ----- DZ7M DBDM7M . TSTEST16 - Zoom Partition ----- 11:48
Command ==>

Partition . . . . . : 0002
Selected . . . . . :

High value of partition . : 16666

```

Figure 25. Zoom Partition panel

Press PF3 to return to the Partition List panel.

What to do next

Specify a primary command in the command field to issue the request and generate the output, or press Enter to access the next panel.

Related reference:

“DB2 HPU interactive component primary commands” on page 213
 Primary commands are used with the DB2 HPU interactive component.

Specifying COPYDDN file parameters

The COPYDDN File Parameters panel is used to specify file parameters for the image copy file.

Procedure

1. Specify the **COPYDDN** command on the General Options panel. The COPYDDN File Parameters panel is displayed, as shown in the following figure:

```

INZA30 ----- DZ7M DBDM7M . TSTEST16 - COPYDDN File Parameters ----- 11:49
Command ==>

Commands : GENERAL PART OPTIONS SELECT UNLDDN JCL

COPYDDN . . . . . N (N - No, nn number, L - Last ic, or DDN)
              (nn is the nth previous version)

If DDN option, please fill the input data set below :

IC . . . . . _ (N - Non-inline, I - Inline, C - Check)
Data set name . . . . . _____
Disposition . . . . . _ (S - Shr, 0 - Old)
Generic unit . . . . . (*)
Volume serial . . . . . _____ - _____ - _____ - _____ (*)

for tape unit
Label . . . . . _____
(*) Used if ImageCopy data set is not cataloged

(PF3 to exit)

```

Figure 26. COPYDDN File Parameters panel

2. Specify values for each of the following fields:

COPYDDN

Displays the COPYDDN File Parameters panel, which is used to specify parameters for the image copy file. This command is available only if a table space was selected.

No No image copy is used as source.

nn The number of the last (most recent) full image copy to be used as source.

Last ic Process the unload job by using the last valid registered full image copy on the object. This option is equivalent to a value of -1 for *nn*.

DDN Process the unload by using the information that is provided on this panel.

IC Specify the image copy type.

Non-inline

Indicates that the copy is not inline.

Inline If the ddname points to an Inline Copy (that is a copy share level change), you must specify Inline. Failure to do so can result in duplicate or missing lines in the output, or failure of the job during the unload process itself.

Check Check must be specified if the copy is Inline or is not known.

Data Set Name

Specify the name of the image copy data set (FlashCopy is not supported).

For partitioned table spaces, to define one input file for each partition, use the &PART variable in the generic data set name, which allows DB2 HPU to process in parallel.

If DB2 HPU was requested to perform processing in parallel for partitioned table spaces, you must specify this processing for all files.

Disp The disposition of the image copy data set. Enter SHR if the image copy data set is to be shared.

Generic unit

Specify the DASD unit name.

Volume serial

Specify the volume serial number. Use an asterisk (*) when the image copy data set is not cataloged.

for tape unit/Label

Specify the label sequence number when the data set resides on tape.

What to do next

Specify a primary command in the command field to issue the request and generate the output, or press Enter to access the next panel.

Related reference:

“DB2 HPU interactive component primary commands” on page 213
Primary commands are used with the DB2 HPU interactive component.

Specifying UNLOAD file parameters

The UNLDDN File Parameters panel is used to specify parameters for allocating the data set that contains the data of the unloaded table spaces or partitions.

Procedure

1. Specify the **UNLDDN** command on the General Options panel. The UNLDDN File Parameters panel is displayed, as shown in the following figure:

```
INZA40 ----- DZ7M DBDM7M . TSTEST16 - UNLDDN File Parameters ----- 11:49
Command ==>

Commands : GENERAL PART COPYDDN OPTIONS SELECT JCL

          Data set with physical UNLOAD of table space or partition

Data set name . . . . . _____
Disposition . . . . . ( _ , _ , _ ) (New/Old/Mod) (Del/Keep/Catlg)
Generic unit . . . . . _____
Volume serial . . . . . _____ - _____ - _____ - _____
Space units . . . . . _____ (BLKS, TRKS, CYLS)
Primary quantity . . . . . _____
Secondary quantity . . . . . _____
Record format . . . . . _____ (F, FB, V, VB)
Record length . . . . . _____
Block size . . . . . _____

SMS
Data class . . . . . _____
Management class . . . . . _____
Storage class . . . . . _____

for tape unit
Label . . . . . _____

( PF3 to exit)
```

Figure 27. Unload File Parameters panel

2. Specify values for each of the following fields:

Data Set Name

Specify the name of the data set that will contain the data of an unloaded table space or partition.

For partitioned table spaces, to define one input file for each partition, use the &PART variable in the generic data set name, which allows DB2 HPU to process in parallel.

If DB2 HPU was requested to perform a processing in parallel for partitioned table spaces, you *must* specify this processing for all files.

Disposition

Specify the disposition of the data set.

New/Old/Mod

Specify the disposition to use as the first parameter.

Del/Keep/Catlg

Specify the disposition to use as the second and third parameters.

Generic unit

Specify a generic unit if DISP=NEW or if the file is not cataloged.

Volume serial

Specify one to five volumes.

Space Units

Specify the volume allocation unit (BLKS, TRKS, or CYLS).

Primary quantity

Specify the primary allocation for the data set.

Secondary quantity

Specify the secondary allocation for the data set.

Record format

Specify the format of the records as F, FB, V, or VB.

Record length

Specify the length of the records. The value that you enter must be less than or equal to 32760.

Block size

Specify the block size if FB or VB is specified for FORMAT. The BLKSIZE value must be less than or equal to 32760 and must be a multiple of LRECL (+4 if V or VB).

SMS Specify the Data Class, Management Class, and Storage Class SMS parameters for allocation of data sets that are managed by SMS. The values that are entered for these parameters will be specified in the DD statement that is generated in the JCL.

for tape unit Label

Specify the tape sequence number if the file will be written to tape.

What to do next

Specify a primary command in the command field to issue the request and generate the output, or press Enter to access the next panel.

Related reference:

“DB2 HPU interactive component primary commands” on page 213
Primary commands are used with the DB2 HPU interactive component.

Specifying options for the SELECT statement

The SELECT Statement Option panels are used to specify options for a SELECT statement.

Procedure

1. Specify the **OPTIONS** command on the General Options panel. The SELECT Statement Option panel (1 of 2) is displayed, as shown in the following figure:

```

INZA501 ---- DZ7M DBDM7M . TSTEST16 - SELECT Statement Option ----- 11:49
Command ==>

Commands      : GENERAL  PART  COPYDDN  SELECT  UNLDDN  JCL

Others options : LOADOPT

NULL . . . . . _ (0 - Off)
or
when NULL . . . . . _ (enter one character or an hexadecimal value)*
and when not NULL . . _ (enter one character or an hexadecimal value)*

NULLID . . . . . _ (Y - Yes, N - No)
NULLPOS . . . . . _ (A - After, B - Before)
PIC sign . . . . . _ (sign or P)
  position . . . . . _ (L - Lead, T - Trail)
  decimal . . . . . _ (, or .)
  mask . . . . . _ (enter ? for the list of authorized
values)
*(case sensitive)

( PF3 to exit, Enter : "Option" 2/2)

```

Figure 28. SELECT Statement Option panel (1 of 2)

2. Specify values for each of the following fields:

- Other options: LoadOPT**
Specify LoadOPT to display an ISPF edit panel that you use to enter load options.
- NULL** Specify whether the null character is to be generated in the output file.
- or when NULL**
Specify the character to be used for columns with NULL value.
- not NULL**
Specify the character to be used for columns that have a value other than NULL.
- NULLID**
Specify whether the output file will contain NULL indicators. This option is considered only for FORMAT USER.
- NULLPOS**
Specify the position of the NULL indicator. It can be specified before or after the data field. This option is considered only for the USER format or it if specified at the select level.
- PIC sign**
Specify the numeric data display format. The default value is a minus sign (-).
 - +** The plus sign (+) is used for positive values, and the minus sign (-) is used for negative values.
 - The minus sign (-) is used for negative values; positive values are preceded by blanks.
 - P** The padding character is used for positive values; negative values are identified by a minus sign (-).
- PIC position**
Specify the position of the sign. The default is Lead.
 - Lead** The sign is placed before the numeric value.

Trail The sign is placed after the numeric value.

PIC decimal

Specify the decimal separator. The default value is a period (.).

. A period is written as the decimal separator.

, A comma is written as the decimal separator.

PIC mask

This optional parameter is used to define the mask. Enter a question mark (?) to display the list of authorized types. See "Selecting mask formats" on page 254 for more information.

3. Press Enter. The second SELECT Statement Option panel is displayed, as shown in the following figure:

```
INZA502 ---- DZ7M DBDM7M . TSTEST16 - SELECT Statement Option ----- 11:49
Command ==>

Commands      : GENERAL  PART   COPYDDN  SELECT  UNLDDN  JCL

Others options : LOADOPT

DATE . . . . . _____
(enter ? for the list of authorized values)
DATEDELIM . . _ (enter one character or an hexadecimal value)*

TIME . . . . . _____
(enter ? for the list of authorized values)
TIMEDELIM . . _ (enter one character or an hexadecimal value)*

TIMESTAMP . . _____
(enter ? for the list of authorized values)
LENGTHBYTE . . _ (Y - Yes, N - No)
LENGTH . . . . _ (R - Real, M - Max)

*(case sensitive)

(PF3 to return to "Option" 1/2)
```

Figure 29. SELECT Statement Option panel (2 of 2)

4. Specify the values that you want for each of the following fields:
 - DATE** Specify the available type to encode dates in FORMAT USER. Specify a question mark (?) to display the list of authorized types.
 - DATEDELIM**
Specify the default delimiter that is to be used in external date representation.
 - TIME** Specify the type to encode times in FORMAT USER. Specify a question mark (?) to display the list of authorized types.
 - TIMEDELIM**
Specify the default delimiter that is to be used in external time representations.
 - TIMESTAMP**
Specify the type to encode timestamps in FORMAT USER. Specify a question mark (?) to display the list of authorized types.
 - LENGTHBYTE**
Specify whether you want the two length bytes for variable-length columns in an options block written to the output data set.

LENGTH

Specify whether the real or maximum length is used for variable-length fields in an options block.

Real The length of the field does not change (value of the two length bytes). The default value is Real.

Max The output field is padded to its maximum length with binary zeros.

The LENGTH keyword is valid only for variable-length fields.

What to do next

Specify a primary command in the command field to issue the request and generate the output, or press Enter to access the next panel.

Related tasks:

“Selecting date formats” on page 251

The Type Values - DATE panel is used to select a date format.

“Selecting time formats” on page 252

The Type Values - TIME panel is used to select a time format.

“Selecting timestamp formats” on page 253

The Type Values - TIMESTAMP panel is used to select a timestamp format.

Related reference:

“DB2 HPU interactive component primary commands” on page 213

Primary commands are used with the DB2 HPU interactive component.

Updating the list of SELECT statements

The SELECT statement list panel is used to update the SELECT statements of a DB2 HPU job.

Procedure

1. Specify the **SELECT** command. The SELECT Statement List panel is displayed, as shown in the following figure.

```
INZA601 ----- DZ7M DBDM7M . TSTEST16 - SELECT Statement List Row 1 to 3 of 3
Command ==>>>                                         Scroll ==>> CSR

Commands : GENERAL      PART      COPYDDN      OPTIONS      UNLDDN      JCL

Line commands: E - Edit, D - Delete, C - Copy

New Select statement ? . . . . _ (Y - Yes)
Use interactives panels . . _ (Y - Yes, N - No, default is Yes)
or
Use a line command on this table of select statements already created.

                                                                    ( PF3 to exit)

Select Select description          Inter. Creator  Table or view
-----
select 01                          Y   IBMUSER  TSTEST16
IBMUSER .TSTEST16                   Y   IBMUSER  TSTEST16
IBMUSER .ZZZZZZZZz                  Y   IBMUSER  TSTEST16
***** Bottom of data *****
```

Figure 30. SELECT Statement List panel

New Select statement ?

Specify Y and press Enter to create a SELECT statement.

If the table space contains several tables, the Table List panel, as shown in Figure 33 on page 232, is displayed. If the table space contains only one table, the SELECT Format panel, as shown in the following figure, is displayed.

Interactive Select

Y Specify Y to display the SELECT Format panel, as shown in the following figure. You can access all the primary commands from this panel.

N If you specify N, the first limited SELECT Format panel is displayed. The commands on this panel are limited; only FORMAT, OUTDDN, LOADDN, and JCL are available. You can use the EDIT command to access ISPF edit mode where you can enter the entire SELECT statement and cut and paste a SYSIN.

```
INZB101E --- DZ7M          .                - SELECT Format ----- 14:51
Command ==>

Commands :  EDIT   OUTDDN   LOADDN   JCL

SELECT Description . . . . .
FORMAT . . . . . 1 (1 - DSNTIAUL, 2 - DELIMITED, 3 - VARIABLE,
                   4 - USER, 5 - EXTERNAL)
DSNTIAUL STRICT . . . . (Y - Yes, N - No)
DELIMITED SEP . . . . (enter one character or an hexadecimal value)*
                   DELIM . . . (enter one character or an hexadecimal value)*
                   NULL DELIM . (Y -Yes, N - No)
VARIABLE . . . . . (E - End, A - All )
LIKE creator . . . . .
   table . . . . .

*(case sensitive)
( PF3 to return to "Select List", Enter : "Format" 2/2)
```

Figure 31. SELECT Format panel (1 of 2)

Press Enter to access the second limited SELECT Format panel, as shown in the following figure.

```

INZB102E --- DZ7M                                - SELECT Format ----- 14:52
Command ==>

Commands : EDIT   OUTDDN   LOADDN   JCL

ORIGINOBID . . . . . (Hexadecimal value)
           or . . . . . (Decimal value)

OUTMAXROWS . . . . .
OUTFREQROWS . . . . .

SCHEME . . . . . (E - EbcDic, A - Ascii, S - aSis, U - Unicode)
CCSID SBCS . . . . .
      MIXED . . . . .
      DBCS . . . . .

OUTEXIT exitname . .
      in . . . . . (1 - ASM, 2 - C, 3 - COBOL2)
                  ( PF3 to return to "Format" 1/2)

```

Figure 32. SELECT Format panel (2 of 2)

The default value is Y.

Selection codes

You can select only one line. Specify one of the following codes:

- S** Use the S selection code to select a statement. The SELECT description panel is displayed for further processing.
- D** Use the D selection code to delete a SELECT statement. A confirmation panel is displayed.
- C** Use the C selection code to copy a statement to a new statement that has the same characteristics and contents as the original statement.

The options you select determine the subsequent panels that are displayed. The following columns are on the SELECT Statement List panel:

S Use this column to enter a selection code.

Select description

This column contains the description of the SELECT statement.

Inter. (Interactive)

- Y** The SELECT statement is created by done using interactive panels.
- N** The SELECT statement is written in freestyle.

Creator

This column contains the name of the creator of the SELECT statement.

Table or view

This column indicates the name of the table or view.

2. Press PF3 to end the current unload JCL set-up.

What to do next

Specify a primary command in the command field to issue the request and generate the output, or press Enter to access the next panel.

Related tasks:

“Selecting tables”

The Table List panel is used to create SELECT statements.

Related reference:

“DB2 HPU interactive component primary commands” on page 213

Primary commands are used with the DB2 HPU interactive component.

Selecting tables

The Table List panel is used to create SELECT statements.

Before you begin

If you Specify non-generic criteria and you do not include a percent sign (%) when listing tables by generic names, a percent sign is automatically added to the end of the field.

About this task

The Table List panel is displayed when you have selected a table space and you Specify the **SELECT** command on the Database and Tablespace Selection panel or when you use the SELECT Statement List panel to create a new SELECT statement and there are several tables in the table space that you selected.

To select a table:

Procedure

Use the S selection code to select one or more tables, and press PF3 to access the SELECT Statement List panel so you can create SELECT statements.

1. If you select only one table, the Table List panel is displayed, as shown in the following figure:

INZA61 ----- DZ7M . - Table List -- Row 1 to 31 of 85
 Command ==> Scroll ==> CSR

Line Commands: S - Select table

Generic: Creator IBMUSER% > (case sensitive)
 Table or view . .

;- > (case sensitive)

Select one or more tables (or views) and press PF3 to create SELECTs

Select	Creator	Table or view	Database	Table space	Type
	IBMUSER	DSN_FUNCTION_TABLE	DBDM7M	DSNRFUNC	T
	IBMUSER	DSN_STATEMNT_TABLE	DBDM7M	DSNRSTAT	T
	IBMUSER	LCLRC	DBDM7M	LCLRC	T
	IBMUSER	TBMASKCHAR	DBDM7M	TBMASKCH	T
	IBMUSER	TESTTH1	DBDM7M	TESTTHC	T
	IBMUSER	TESTTH2	DBDM7M	TESTTHC	T
	IBMUSER	TESTTH3	DBDM7M	TESTTHC	T
	IBMUSER	TESTTH01	DBDM7M	TESTTH01	T
	IBMUSER	TSDEC	DBDM7M	TSDEC	T
	IBMUSER	TBEXT	DBDM7M	TSEXT	T
	IBMUSER	TBFLOAT	DBDM7M	TSFLOAT	T
	IBMUSER	TBID3121	DBDM7M	TSID3121	T
	IBMUSER	TBID3610	DBDM7M	TSID3610	T
	IBMUSER	EMP_PHOTO_RESUME	DBDM7M	TSLOBTH1	T
	IBMUSER	TBMASK	DBDM7M	TSMASK	T
	IBMUSER	TSPART1	DBDM7M	TSPART1	T
	IBMUSER	TBPAR10	DBDM7M	TSPAR10	T
	IBMUSER	TBPAR100	DBDM7M	TSPAR100	T
	IBMUSER	TBPAR250	DBDM7M	TSPAR250	T
	IBMUSER	TBPAR254	DBDM7M	TSPAR254	T
	IBMUSER	TBPB5930	DBDM7M	TSPB5930	T
	IBMUSER	TBSCA	DBDM7M	TSSCA	T
	IBMUSER	TBSCA2	DBDM7M	TSSCA2	T
	IBMUSER	TBSIM01	DBDM7M	TSSIM01	T
	IBMUSER	VUESIM01	DBDM7M	TSSIM01	V
	IBMUSER	TSTEST16	DBDM7M	TSTEST16	T
	IBMUSER	TB2739	DBDM7M	TS2739	T
	IBMUSER	TS2904	DBDM7M	TS2904	T
	IBMUSER	TS2904B	DBDM7M	TS2904B	T
	IBMUSER	TS2922	DBDM7M	TS2922	T
	IBMUSER	TB2934	DBDM7M	TS2934	T

```

+----- UGCREA+0 -----+
| ISPEXPND                               Line 1 of 2 |
| Command ==>                               Scroll ==> PAGE |
| IBMUSER%                               |
|-----|
|-----|
|-----|
|-----|
|-----|
|-----|
|-----|
+-----+
  
```

Figure 33. Table List panel

The Table List panel contains the following columns:

S The selection code; enter one or more selection codes in this column.

Creator

This column contains the name of the creator of the table.

Table or view

This column contains the name of the table or of the view.

Database

This column contains the name of the database.

Table space

This column contains the name of the table space.

Type This column contains the type (table or view).

- If several tables are selected, the SELECT Statement List panel is displayed.

```

INZA601 ----- DZ7M DBDM7M . TSTEST16 - SELECT Statement List Row 1 to 3 of 3
Command ==> Scroll ==> CSR

Commands : GENERAL PART COPYDDN OPTIONS UNLDDN JCL

Line commands: E - Edit, D - Delete, C - Copy

New Select statement ? . . . . _ (Y - Yes)
Use interactives panels . . _ (Y - Yes, N - No, default is Yes)
or
Use a line command on this table of select statements already created.

                                                                    ( PF3 to exit)

Select Select description          Inter. Creator Table or view
-----
select 01                          Y   IBMUSER TSTEST16
IBMUSER .TSTEST16                   Y   IBMUSER TSTEST16
IBMUSER .ZZZZZZZZz                   Y   IBMUSER TSTEST16
***** Bottom of data *****

```

Figure 34. SELECT Statement List panel

What to do next

Press PF3 to exit this panel.

Related tasks:

“Selecting time formats” on page 252

The Type Values - TIME panel is used to select a time format.

“Updating the list of SELECT statements” on page 228

The SELECT statement list panel is used to update the SELECT statements of a DB2 HPU job.

“Starting DB2 HPU interactive component by using the DB2 Tools Launchpad” on page 217

The DB2 HPU interactive component can be started by using the DB2 Tools Launchpad.

Processing long names

DB2 HPU accepts long names for tables and columns. You can view and update the names by using the Complete table name panel.

Procedure

- Position the cursor on the corresponding field, and press PF5. PF5 is available only when a table name is either displayed or can be entered on the panel; therefore, this option is not available on every panel. The Complete table name

panel is displayed, as shown in the following figure:

```

INZP041 ----- *** Complete table name *** ----- 11:50
Command ==>

Short name : DEPARTMENT_PHONE_N
Long name  : DEPARTMENT_PHONE_NUMBER
  
```

Figure 35. Complete table name panel

2. Press PF3 to save your modifications, or issue the CANCEL command to cancel.

Specifying the format of the SELECT statement

The SELECT Format panels are used to specify the output format for a SELECT statement.

Procedure

1. Specify the **FORMAT** command to display the first SELECT Format panel.

```

INZB1011 --- DZ7M IBMUSER . TBEXT          - SELECT Format ----- 11:56
Command ==>

Commands : COLUMNS  WHERE  ORDERBY  OUTDDN  FUSER  LOADDN  JCL

Other Option : PART

SELECT Description . . IBMUSER .TBEXT
PART . . . . . * (L - List, * - ALL, nnnn - partition number)
FORMAT . . . . . 1 (1 - DSNTIAUL, 2 - DELIMITED, 3 - VARIABLE,
                   4 - USER, 5 - EXTERNAL)
DSNTIAUL STRICT . . . _ (Y - Yes, N - No)
DELIMITED SEP . . . _ (enter one character or an hexadecimal value)*
  DELIM . . . _ (enter one character or an hexadecimal value)*
  NULL DELIM. . . _ (Y -Yes, N - No)
VARIABLE. . . . . _ (E - End, A - All )
LIKE creator . . . .
  table . . . .

*(case sensitive)
( PF3 to return to "Select List", Enter : "Format" 2/2)
  
```

Figure 36. SELECT Format panel (1 of 2)

2. Specify the values for the following fields in the first SELECT Format panel:

Other Option : PART

Use the PART option to access the Partition Selection panels to select a partition for the current SELECT statement. This SELECT format (list of partitions) panel is displayed when you specify PART in the **Other Option** field.

Use this SELECT format panel to select one or more of the displayed partitions.

```

INZB12 ----- DZ7M IBMUSER . TSTEST16          - SELECT Format -- Row 2 of 14
Command ==>                                         Scroll ==> CSR

Line comands: S - Select partition, U - Unselect, Z - Zoom

SELECT Description . . IBMUSER .TSTEST16
                                                    ( PF3 to return to "Format")

S Part  Selected High value of partition
-----
- ALL   YES
- 0001      8333
- 0002     16666
- 0003     24999
- 0004     33333
- 0005     41666
- 0006     99999
***** Bottom of data *****

```

Figure 37. SELECT Format (partition list) panel

The selection codes for the Partition List panel are:

- S** Specify S next to any partition you want to select.
- U** Specify U next to any partition to undo its selection.
- Z** Specify Z next to any partition to zoom in on a partition and to display on a single panel all information concerning that partition. An example is shown in Figure 25 on page 222.

Requirement: You must select at least one partition.

The columns on the SELECT format (partition list) panel are:

- S** Use this column to specify the partitions to be selected or deselected. Enter a selection code in this column.
- Part** This column indicates the available partitions. Only the partitions that were selected on the Partition list panel, Figure 24 on page 221, are displayed in this panel.

To select all partitions, specify the S selection code beside **ALL** in the Part column.

To select a partition, specify the S beside a partition number.

To deselect a partition, specify the U selection code beside a partition number.

Requirement: You must select at least one partition.

Selected

If YES is specified, the partition was selected. This field is displayed after you select a partition and press Enter.

High value of partition

Shows the first 57 characters of the LIMITKEY column of the SYSIBM.SYSTABLEPART table.

Use the PF3 key to return to the first SELECT format panel, as shown in Figure 36 on page 234.

SELECT Description

Type a short description of the SELECT statement. The description is

displayed in the Select description column of the SELECT Statement List panel, as shown in Figure 30 on page 228.

PART Specify the following information about the partitions of a table space:

- L** Indicates that several partitions have been selected by using the **PART** command. Selection code **L** is not used to display the list of partitions.
- *** Selects all partitions of a table space.
- nnn** Indicates the number of a single partition of the table space.

FORMAT

Specify the format of the output file:

1 : DSNTIAUL

Specifies that the file that is produced must be in the format required by DSNTIAUL.

2 : DELIMITED

Specifies that the output file can be processed by your personal computer.

3 : VARIABLE

Specifies that the output file must be compatible with the DB2 LOAD utility.

4 : USER

Specifies that there is output in USER format.

DSNTIAUL STRICT

Yes Specify Yes to obtain constant chains (if any are present in the SELECT statements) in variable format (VARCHAR). You can use only DSNTIAUL STRICT when you select option 1, FORMAT DSNTIAUL.

No Specify No if you do not want to obtain constant chains in variable format.

DELIMITED

Specify that the format of the output data set is a comma-separated value file which corresponds to a .csv file that can be processed by your personal computer. You can specify only DELIMITED when you select option 2, FORMAT DELIMITED.

SEP

Use this field to specify the character to be used to separate fields in the output data set.

DELIM

Enter the delimiter character to be used to enclose CHAR, VARCHAR, GRAPHIC, and VARGRAPHIC fields in the output data set.

NULL DELIM

Yes Specifies that the delimiter character will be used at the beginning and the end of a field containing nulls.

No Specifies that the delimiter character will not be used at the beginning and the end of a field that contains nulls.

VARIABLE

You can specify only VARIABLE when you select option 3, FORMAT VARIABLE. If you select option 3, you must enter a value in this field.

END Specify that the characteristics and the sequence of fields of the generated file are the same as the selected columns.

ALL Specify that the variable columns are to be written according to their lengths.

LIKE creator

Specify the user ID of the creator. You can specify only LIKE creator when you select option 1, FORMAT DSNTIAUL or option 3, FORMAT VARIABLE.

LIKE table

Specify either the short or long table name that indicates the format to be used when creating the output file. You can specify only LIKE table when you select option 1, FORMAT DSNTIAUL or option 3, FORMAT VARIABLE.

3. Press Enter to display the second SELECT format panel or press PF3 to return to the Select List. The second SELECT format panel is displayed, as shown in the following figure:

```
INZB102I --- DZ7M IBMUSER . TSTEST16          - SELECT Format ----- 11:59
Command ==>

Commands : COLUMNS  WHERE  ORDERBY  OUTDDN  FUSER  LOADDN  JCL

Other Option : PART

ORIGINOBID . . . . . ____ (Hexadecimal value)
              or . . . . . ____ (Decimal value)

OUTMAXROWS . . . . . _____
OUTFREQROWS . . . . . _____

SCHEME . . . . . _ (E - EbcDic, A - Ascii, S - aSis, U - Unicode)
CCSID SBCS . . . . . _____
      MIXED . . . . . _____
      DBCS . . . . . _____

OUTEXIT exitname . . _____
      in . . . . . _ (1 - ASM, 2 - C, 3 - COBOL2)

                                     ( PF3 to return to "Format" 1/2)
```

Figure 38. SELECT format panel (2 of 2)

4. Specify the following values for the second SELECT format panel:

ORIGINOBID

Specify a value if the OBID table in the image copy is not equal to the OBID read in the catalog. A discrepancy can occur, for example, for an image copy of a table that was dropped and re-created with a new OBID.

OBID is the hexadecimal value of the OBID of the table in the image copy.

Use the appropriate line to enter a hexadecimal or a decimal value for ORIGINOBID.

OUTMAXROWS

Specify the maximum number of rows to select.

OUTFREQROWS

Specify the unload sampling frequency. One row out of every n rows is written into the OUTDDN data set.

SCHEME

This parameter is optional; use it to specify the format in which the data is unloaded.

E EBCDIC
A ASCII
S ASIS
U Unicode

CCSID

Specify as many as three valid optional code character set identifiers for the unloaded data: SBCS, MIXED, and DBCS values. If any of the fields are omitted or are specified as 0, the corresponding CCSID is assumed to be the same as the installation default CCSID.

You can load the FIELDPROC and EDITPROC directly from the DSNEXIT that is specified in the PARMLIB without allocating it in STEPLIB.

Attention: If the TRANSLAT parameter in the PARMLIB is set to inhibit data conversion, you can specify a conversion parameter (ASCII, EBCDIC, ASIS, or CCSID) in the SYSIN to override this inhibition.

If the unload format that is specified either in the SYSIN or in the PARMLIB by using the UNLSCHEM parameter is not identical to the system's EBCDIC format, all constants that are specified in SYSIN will be translated to the unload format.

OUTEXIT *exitname* in *exit_language*

Use this field to specify the name and the language of the exit that handles the rows during the unload processing.

ASM Assembler language (default)
C C language
COBOL2
COBOL/2 language

5. Press Enter to accept the values, or press PF3 key to return to the first SELECT format panel.

Related reference:

"DB2 HPU interactive component primary commands" on page 213
Primary commands are used with the DB2 HPU interactive component.

Selecting columns for the SELECT statement

The SELECT Columns panel is used to select columns for the SELECT statement.

Procedure

Specify the **COLUMNS** command. The SELECT Columns panel is displayed, as shown in the following figure:

```

INZB201 ---- DZ7M IBMUSER . TSTEST16          - SELECT Columns Row 1 from 5
Command ==>                                     Scroll ==> CSR

Commands :  FORMAT   WHERE   ORDERBY   OUTDDN   FUSER   LOADDN   JCL

Other Option:  SELECTALL

Line commands: E - Edit, D - Delete

SELECT Description . . IBMUSER .TSTEST16
                                     ( PF3 to return to "Select List")

      Pos in <----- Columns description ----->
Sel Select Name/Value/Expr.  Trunc L/C/E Type      Length Scale Pos in table
-----
- 1 COL1                      C  INTEGER          <=== NEW
- 2 COL2                      C  CHAR              50
- 3 AAAAAAA                   L  CHAR              15
- 4 COL4                      C  VARCHAR           50
- 5 COL3                      C  CHAR              50
***** Bottom of data *****

```

Figure 39. SELECT Columns panel

The SELECT columns panel contains the following fields. You can edit the **SELECT Description**, **Sel**, and **Pos in Select** fields.

SELECT Description

Type a short description of the SELECT statement. The description is displayed in the Select description column of the SELECT Statement List panel, as shown in Figure 30 on page 228.

Sel Specify one of the following options:

- E** Edits the line. When you select E, the Edit Expression panel is displayed, as shown in the following figure. Use the Edit Expression panel to display the literal or the expression of a SELECT statement.

```

INZB21 ---- DZ7M IBMUSER . TSTEST16          - Edit Expression ----- 12:00
Command ==>                                     Scroll ==> CSR

Column Type . . . L          (L - Literal, E - Expression)
Pos in Select : : 3
Type . . . . . CHAR        (? will display a list of authorized values)
Length . . . . . 15
Scale . . . . .
Value . . . . . AAAAAAA

                                     ( PF3 to return to "Select Columns")

```

Figure 40. Edit Expression panel

Use the PF3 key to return to the SELECT columns panel.

- D** Deletes the line.

Pos in Select

Specify the position of the column in the SELECT statement.

Name/Value/Expr.

Indicates the name of the column, the beginning of the literal value, or the expression.

- Trunc** Indicates that the literal or column name is truncated on this panel.
- L/C/E** Indicates that the line is related to a literal (L), to a column (C), or to an expression (E).
- Type** Indicates the type of the column or the literal.
- Length**
Indicates the length of the column or the literal.
- Scale** Indicates the decimal part of the column or the literal.
- Pos in table**
Indicates the position of the column in the table. Literals are indicated by a 0 in this column.
- Related reference:**
"DB2 HPU interactive component primary commands" on page 213
Primary commands are used with the DB2 HPU interactive component.

Sorting columns of a SELECT statement

The SELECT ORDER BY panel is used to specify the order in which each column in the SELECT statement is processed for the output data set and whether to sort the column in ascending or descending order.

Procedure

Specify the **ORDER BY** command. The SELECT Order By panel is displayed, as shown in the following figure:

The SELECT ORDER BY panel contains the following fields. You can edit the

```

INZB40 ---- DZ7M IBMUSER . TSTEST16          - SELECT Order By Row 1 from 5
Command ==>                               Scroll ==> CSR

Commands :  FORMAT   COLUMNS   WHERE   OUTDDN   FUSER   LOADDN   JCL
           *

Line commands: D - Delete column from the Order-By clause.

SELECT Description . . IBMUSER .TSTEST16
                                     ( PF3 to return to "Select List")

<----- Columns description -----> Pos in Pos in
Select Name      Type      Length Scale Select Order (A/D)
-----
- COL4           VARCHAR   50      4 1  A
- COL2           CHAR     50      2 2  D
- COL1           INTEGER  50      1 3  A
- COL3           CHAR     50      5 4  -
***** Bottom of data *****

```

Figure 41. SELECT Order By panel

SELECT description, **Select**, **Pos in order**, and **(A/D)** fields.

SELECT Description

Type a short description of the SELECT statement. The description is displayed in the Select description column of the SELECT Statement List panel, as shown in Figure 30 on page 228.

Select Specify D in this column to delete the column.

Name Indicates the name of the column.

Type Indicates the type of the column.

Length

Indicates the total length of the column. This field is only applicable for the CHAR, VARCHAR, LONGVAR, DECIMAL, FLOAT, GRAPHIC, VARG, and LONGVARG column types.

Scale Indicates the length of the decimal part of the column. This field is authorized for columns of type DECIMAL.

Pos in Select

Indicates the position of the column in the SELECT statement.

Pos in Order

Specify the position of the column in creating the output data set.

A/D Specify either A to sort in ascending order or D to sort in descending order.

Related reference:

“DB2 HPU interactive component primary commands” on page 213
Primary commands are used with the DB2 HPU interactive component.

Displaying and modifying the USER format of the SELECT statement

The Select user format panels are used to update the USER formats.

Procedure

1. Specify the **FUSER** command. The first Select User Format panel is displayed, as shown in the following figure:

```
INZB501 -- DZ7M IBMUSER . TSTEST16          - Select User Forma Row 1 from 5
Command ==>                                Scroll ==> CSR

Commands :  FORMAT      COLUMNS  WHERE    ORDERBY  OUTDDN   LOADDN   JCL
            *           *                *

Line commands: Z - Zoom

SELECT Description . . IBMUSER .TSTEST16
                                                    ( PF3 to return to "Select List")

<----- Columns description -----> <----- Field description ----->
                                                    (PF11) -->
S Name          Type      Length Scale Type      Length Scale Pad Just
-----
- COL1          INTEGER
- COL2          CHAR       50
- AAAAAAAA     CHAR       15
- COL4          VARCHAR    50
- COL3          CHAR       50
***** Bottom of data *****
```

Figure 42. Select User Format panel (1 of 4)

The following fields are available on all four Select User Format panels. You can edit the **SELECT description** and the **S** fields.

SELECT Description

Type a short description of the SELECT statement. The description is displayed in the Select description column of the SELECT Statement List panel.

S Specify the Z line command in this column to specify the columns that you want to zoom. The zoom function displays two panels that contain all the information about a single column. The following figure shows the first User Format Zoom panel:

```

INZB51 ---- DZ7M IBMUSER . TSTEST16          - User Format Zoom ----- 12:02
Command ==>

Columns description
Name . . . . : COL2                Pos in table . . : 2
Type . . . . : CHAR                50          Pos in Select . . : 2
Null/def . . : N                  WD          Pos in ORDER BY . : 2      D

Field description
TYPE . . . . . _____ ( ? will display a list of authorized values)
  Length . . _____
  Scale . . _____
PADDING . . . . . _____ (one character or an hexadecimal value)*
JUST . . . . . _____ (R - Right, L - Left)
DELIM . . . . . _____ (one character or an hexadecimal value)*
LENGTHBYTE . . . . . _____ (Y - Yes, N - No)
LENGTH . . . . . _____ (R - Real, M - Max)
CCSID . . . . . _____ (number, UTF8, UTF16)

*(case sensitive)
( PF3 to return to "User Format", Enter : "Zoom Panel" 2/2 )

```

Figure 43. User Format Zoom panel (1 of 2)

Use the PF3 key to return to the Select User Format panel, or press Enter to display the second User Format Zoom panel, as shown in the following figure:

```

INZB52 ---- DZ7M IBMUSER . TSTEST16          - User Format Zoom ----- 12:03
Command ==>

Columns description
Name . . . . : COL2                Pos in table . . : 2
Type . . . . : CHAR                50          Pos in Select . . : 2
Null/def . . : N                  WD          Pos in ORDER BY . : 2      D

Field description
PIC sign . . . . . _____ (sign, P - Padded)
  position . . . . . _____ (L - Lead, T - Trail)
  decimal . . . . . _____ (, or .)
  mask . . . . . _____ (enter ? for the list of authorized values)

NULL . . . . . _____ (0 - Off ) (NULL Off overrides NULLID)
  or when NULL . . . . . _____ (one character or hexadecimal value)*
  not NULL . . . . . _____ (one character or hexadecimal value)*

NULLID . . . . . _____ (Y - Yes, N - No)

*(case sensitive)
( PF3 to return to "Zoom Panel" 1/2 )

```

Figure 44. User Format Zoom panel (2 of 2)

Use the PF3 key to return to the Select User Format panel.

Name Indicates the name of the column.

Type Indicates the type of the column.

Len. Indicates the length of the column.

Scale Indicates the decimal part of the column.

2. Edit any of the following fields on the first panel and press Enter.

- Type** Specify the type of the column after the conversion.
- Len.** Specify the total length of the column. This field is authorized for columns of type CHAR, DECIMAL, ..., and its value must be less than the value shown in Length.
- Scale** Specify the length of the decimal part of the column. This field is authorized for columns of type DECIMAL, ..., and its value must be less than the value shown in Length.
- Pad** Specify the padding character in hexadecimal.
- Just** Specify L for left justification or R for right justification.

The second Select User Format panel is displayed, as shown in the following figure:

```

INZB502 -- DZ7M IBMUSER . TSTEST16          - Select User Forma Row 1 from 5
Command ==>                               Scroll ==> CSR

Commands :  FORMAT   COLUMNS   WHERE   ORDERBY   OUTDDN   LOADDN   JCL
           *         *         *         *         *         *         *

Line commands: Z - Zoom

SELECT Description . . IBMUSER .TSTEST16
                                     ( PF3 to return to "Select List")

<----- Columns description -----> <---- Field description -->
<-- (PF10) (PF11) -->
S Name          Type      Length Scale Nul/def Delim Lengthbyte Length
-----
COL1            INTEGER          N
COL2            CHAR          50  N WD
AAAAAAAAA      CHAR          15  N
COL4            VARCHAR         50  Y WD
COL3            CHAR          50  N WD
***** Bottom of data *****

```

Figure 45. Select User Format panel (2 of 4)

The following field is available only on the second Select User Format panel. You cannot edit this field.

Nul/def

Indicates the Null option of the column.

N The Null is not authorized.

Y The Null is authorized.

WD With Default.

3. Edit any of the following fields on the second Select User Format panel and press Enter.

Delim Specify the delimiter character at the beginning and the end of a field.

Lengthbyte

Specify the length byte.

Length

Specify whether the real or maximum length is to be used for fields of variable length.

The third Select User Format panel is displayed, as shown in the following figure:

```

INZB503 -- DZ7M IBMUSER . TSTEST16          - Select User Forma Row 1 from 5
Command ==>                               Scroll ==> CSR

Commands :  FORMAT      COLUMNS    WHERE      ORDERBY    OUTDDN    LOADDN    JCL
           *           *           *           *

Line commands: Z - Zoom

SELECT Description . . IBMUSER .TSTEST16
                                     ( PF3 to return to "Select List")

<----- Columns description -----> <----- Field description ----->
                                     <-- (PF10) (PF11) -->
S Name          Type      Length Scale Sign Pos Dec Mask CCSID
-----
- COL1          INTEGER
- COL2          CHAR       50      -   -   -   -   -
- AAAAAAAA     CHAR       15      -   -   -   -   -
- COL4          VARCHAR    50      -   -   -   -   -
- COL3          CHAR       50      -   -   -   -   -
***** Bottom of data *****

```

Figure 46. Select User Format panel (3 of 4)

- Edit any of the following fields on the third Select User Format panel and press Enter.

Sign Specify the numeric data display format.

Pos Specify the position of the sign.

Dec Specify the decimal separator.

Mask Specify the mask value.

CCSID

You can specify up to three optional identifiers of the code character set code for the unloaded data: SBCS, MIXED, and DBCS values. If any field is omitted or is specified as 0, the corresponding CCSID is assumed to be the same as the installation default CCSID.

You can load the FIELDPROC and EDITPROC directly from the DSNEXIT that is specified in the PARMLIB without allocating it in STEPLIB.

Attention: If the TRANSLAT parameter in the PARMLIB is set to inhibit data conversion, you can specify a conversion parameter (ASCII, EBCDIC, ASIS, or CCSID) in the SYSIN to override this inhibition.

If the unload format that is specified either in the SYSIN or in the PARMLIB by using the UNLSCHEM parameter is not identical to the EBCDIC format of the system, all constants that are specified in SYSIN are translated to the unload format.

The fourth Select User Format panel is displayed, as shown in the following figure:

```

INZB504 -- DZ7M IBMUSER . TSTEST16          - Select User Forma Row 1 from 5
Command ==>                               Scroll ==> CSR

Commands :  FORMAT   COLUMNS   WHERE   ORDERBY   OUTDDN   LOADDN   JCL
           *         *         *         *         *         *         *

Line commands: Z - Zoom

SELECT Description . . IBMUSER .TSTEST16
                                     ( PF3 to return to "Select List")

<----- Columns description -----> <----- Field description ----->
                                     <-- (PF10)
S Name          Type      Length Scale Null When Null Not Null  Null ID
-----
- COL1          INTEGER
- COL2          CHAR       50      -      -      -      -
- AAAAAAAAAA    CHAR       15      -      -      -      -
- COL4          VARCHAR    50      -      -      -      -
- COL3          CHAR       50      -      -      -      -
***** Bottom of data *****

```

Figure 47. Select User Format panel (4 of 4)

- Edit any of the following fields on the fourth Select User Format panel, and press Enter.
 - Null** Specify whether the null character is to be generated in the output file.
 - Not Null** Specify the character to be used for columns having a value other than Null.
 - Null ID** Specify whether the output file will contain Null indicators. This option is available only for the USER format or, if specified, at the SELECT level.

Related reference:
 "DB2 HPU interactive component primary commands" on page 213
 Primary commands are used with the DB2 HPU interactive component.

Displaying the list of OUTDDN files

The OUTDDN list panel is used to display the list of OUTDDN files.

Procedure

- Specify the **OUTDDN** command. The OUTDDN List panel is displayed, as shown in the following figure:

```

INZB601I ---- DZ7M IBMUSER . TSTEST16          - OUTDDN List ---- Row 1 of 2
Command ==>                                     Scroll ==> CSR

Commands :  FORMAT   COLUMNS   WHERE   ORDERBY   FUSER   LOADDN   JCL
           *         *         *         *         *         *         *

Line commands: E - Edit, D - Delete, C - Copy

SELECT Description . . IBMUSER .TSTEST16

Select a data set or enter a new data set name :

                                                    ( PF3 to return to "Select List")

S Data set name                               Disposition
-----
- _____ <=== NEW
- IBMUSER.UNLOAD01                            ( N , C , D )
- IBMUSER.UNLOAD02                            ( N , C , D )

***** Bottom of data *****

```

Figure 48. OUTDDN List panel

You can use the following selection codes on the OUTDDN List panel:
The OUTDDN List panel contains the following fields:

- S** Use this column to specify the partitions to be selected or deselected. You can enter one of the following selection codes in this column:
 - S** Use the S selection code to specify values for output data sets.
 - D** Deletes a file.
 - C** Copies a file to a new file.

Data Set Name

Indicates the name of the data set that will contain the result of the SELECT statement.

For partitioned table spaces, to define one output file for each partition, use the &PART variable in the generic data set name, which allows DB2 HPU to process in parallel.

If DB2 HPU was requested to perform parallel processing for partitioned table spaces, you must specify this processing for all files.

Disp Indicates the disposition of the OUTDDN files.

New/Old/Mod

Indicates the disposition to use as the first parameter.

Del/Keep/Catlg

Indicates the disposition to use as the second and third parameters.

2. Specify the **S** selection code in the **S** field. The OUTDDN File panel is displayed, as shown in the following figure:

```

INZB61 ----- DZ7M IBMUSER . TSTEST16          - OUTDDN File ----- 12:04
Command ==>

                                Data set with formatted result of SELECT

Data set name . . . . : IBMUSER.UNLOAD03
Disposition . . . . ( _ , _ , _ ) (New/Old/Mod) (Del/Keep/Catlg)
Generic unit . . . . _____
Volume serial . . . . _____ - _____ - _____ - _____
Space units . . . . _____ (BLKS, TRKS, CYLS)
Primary quantity . . . . _____
Secondary quantity . . . . _____
Record format . . . . _ (F, FB, V, VB)
Record length . . . . _____
Block size . . . . _____

SMS
Data class . . . . . _____
Management class . . . . _____
Storage class . . . . _____

for tape unit
Label . . . . . _____

                                ( PF3 to return to "OUTDDN List")

```

Figure 49. OUTDDN File panel

3. Edit the following fields:

Data Set Name

Indicates the name of the OUTDDN data set.

For partitioned table spaces, to define one output file for each partition, use the &PART variable in the generic data set name, which allows DB2 HPU to process in parallel.

If DB2 HPU was requested to perform a processing in parallel for partitioned table spaces, you *must* specify this processing for all files.

Disposition

Specify whether the OUTDDN data set is shared.

New/Old/Mod

Specify the disposition to use as the first parameter.

Del/Keep/Catlg

Specify the disposition to use as the second and third parameters.

Generic unit

Specify the DASD unit name.

Volume serial

Specify the volume serial number.

Space units

Specify the volume allocation unit (BLKS, TRKS, or CYLS).

Primary quantity

Specify the primary allocation for the data set.

Secondary quantity

Specify the secondary allocation for the data set.

Record format

Specify the format of the records as F, FB, V, or VB.

Record length

Specify the length of the records with a value less than or equal to 32760.

Block size

If FB or VB was specified in the FORMAT field, you must specify the block size. The BLKSIZE value must be less than or equal to 32760 and it must be a multiple of LRECL (+4 if V or VB).

SMS Specify the Data Class, Management Class, and Storage Class parameters for allocation data sets that are managed by SMS. The values that are entered for these parameters are in the DD statement that is generated in the JCL.

for tape unit/Label

Specify the tape sequence number if the file is to be written to tape.

Related reference:

“DB2 HPU interactive component primary commands” on page 213
 Primary commands are used with the DB2 HPU interactive component.

Displaying and modifying LOADDDN file parameters

The LOADDDN file panel is used to display and modify LOADDDN file parameters.

Procedure

1. Specify the **LOADDDN** command. The LOADDDN File panel is displayed, as shown in the following figure:

```

INZB70I ----- DZ7M IBMUSER . TSTEST16          - LOADDDN File ----- 12:05
Command ==>

Commands :  FORMAT   COLUMNS   WHERE   ORDERBY   OUTDDN   FUSER   JCL
            *         *                   *         *
Other Option :  LOADOPT

                          Generated SYSIN LOAD data set

SELECT Description . . IBMUSER .TSTEST16
-----
SYSOUT class . . . . . _
or
Data set name . . . . . _____
Member name for PDS . . _____
Disposition . . . . . ( _ , _ , _ ) (New/01d/Mod) (Del/Keep/Cat1g)
Generic unit . . . . . _____
Volume serial . . . . . _____ - _____ - _____ - _____

for tape unit
Label . . . . . _____

                                     ( PF3 to return to "SELECT List")

```

Figure 50. LOADDDN File panel

2. Edit the following fields:

Other options: LoadOPT

Specify LoadOPT to display an ISPF edit panel that you use to enter load options.

SELECT Description

Specify a short description of the SELECT statement. The description is displayed in the Select description column of the SELECT Statement List panel.

SYSOUT class or Data Set Name

Specify the SYSOUT class or the data set name for the LOADDDN file. If you enter a value in both fields, the value entered in the SYSOUT class field is used.

Member name for PDS

Specify the PDS member name.

Disp Specify the disposition and whether the LOADDDN data set is to be shared.

Generic unit

Specify the DASD unit name.

Volume serial

Specify the volume serial number.

for tape unit Label

Specify the tape sequence number, if the file will be written to tape.

Related reference:

“DB2 HPU interactive component primary commands” on page 213
Primary commands are used with the DB2 HPU interactive component.

Generating JCL to run a DB2 HPU job

You can generate JCL, edit JCL, submit JCL, save JCL, and enter job card parameters for a DB2 HPU job by invoking the JCL command.

Procedure

1. Specify the **JCL** command.

The EDIT, SUBmit, SaveJCL, JobCard window is displayed when you enter the JCL command.

```

INZA10          ----- General Options ----- 12:05
Command ==>

Commands : PART    COPYDDN    OPTIONS    SELECT    UNLDDN    JCL
                *
DB2 system name . : DZ7M      Object name . : DBDM7M . T
                +-----+
                |  EDIT  |
                | SUBmit |
                | SaveJCL|
                | JobCard|
                +-----+

Utility ID . . . HPU
Part . . . . . * (L - list, * - all, nnn - partition number)
                0  partition(s) selected from 6 total partitions

DB2 . . . . . (Y - Yes, N - No, F - Force)
LOCK . . . . . (Y - Yes, N - No)
QUIESCE . . . . (Y - Yes, N - No)
QUIESCECAT . . . (Y - Yes, N - No)
UNLMAXROWS . . .
UNLFREQROWS . .

(PF3 to exit)

```

Figure 51. EDIT, SUBmit, SaveJCL, JobCard window

2. Position the cursor beside one of the following options and press Enter.

EDIT Select this option to generate and edit JCL. An ISPF edit session is opened on the generated JCL. You can review it, modify it, and resubmit it.

SUBmit
Select this option to generate and submit JCL. The job is generated and submitted.

SaveJCL
Select this option to generate and save JCL.

JobCard
Select this option to update the job card that is used to generate JCL.

Related tasks:

“Saving JCL into a file”

You can save the generated JCL into a file.

“Updating the job card” on page 251

This option is used to update the job card that is used for JCL generation.

Related reference:

“DB2 HPU interactive component primary commands” on page 213

Primary commands are used with the DB2 HPU interactive component.

Saving JCL into a file

You can save the generated JCL into a file.

About this task

To save generated JCL into a file:

Procedure

1. Specify the JCL command.
2. Select the SAVE JCL option on the EDIT, SUBmit, SaveJCL, JobCard window. The SAVE JCL panel is displayed, as shown in the following figure:

```
INZA10          ----- General Options ----- 12:06
C +----- Save JCL -----+
C | INZP055
C | Command ==>
D |
D | Dsname . . _____
D | Member . . _____
+-----+
Utility ID . . . HPU
Part . . . . . * (L - list, * - all, nnn - partition number)
                0 partition(s) selected from 6 total partitions

DB2 . . . . . (Y - Yes, N - No, F - Force)
LOCK . . . . . (Y - Yes, N - No)
QUIESCE . . . . (Y - Yes, N - No)
QUIESCECAT . . . (Y - Yes, N - No)
UNLMAXROWS . . .
UNLREQROWS . . .

(PF3 to exit)
```

Figure 52. SAVE JCL panel

3. Edit the following fields:

DS name

Specify the name of an existing PDS where you want to save the JCL. If the name is entered between single quotation marks, the TSO prefix is not added.

Member

Specify the member name. If this member exists, it is by the new one.

Related tasks:

“Generating JCL to run a DB2 HPU job” on page 249

You can generate JCL, edit JCL, submit JCL, save JCL, and enter job card parameters for a DB2 HPU job by invoking the JCL command.

Updating the job card

This option is used to update the job card that is used for JCL generation.

Procedure

1. Specify the **JCL** command.
2. Select the JobCard option on the EDIT, SUBmit, SaveJCL, JobCard window. The User Job Card panel is displayed, as shown in the following figure:

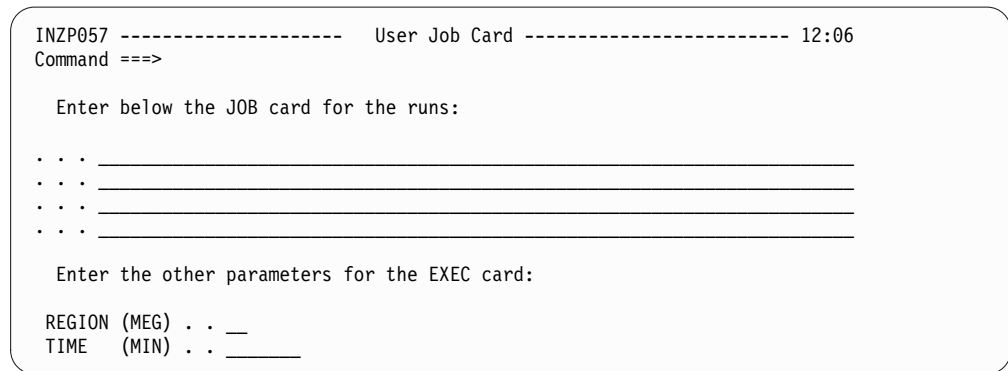


Figure 53. User Job Card panel

3. Edit the following fields:

Enter below the JOB card for the runs:

Specify 1 - 4 lines for the job card. The syntax is identical to the standard job card syntax.

REGION (MEG)

Specify the maximum memory size (in MB) for the unload step.

TIME (MIN)

Specify the CPU time consumption limit (in minutes) for the unload step.

Related tasks:

“Generating JCL to run a DB2 HPU job” on page 249

You can generate JCL, edit JCL, submit JCL, save JCL, and enter job card parameters for a DB2 HPU job by invoking the JCL command.

Selecting date formats

The Type Values - DATE panel is used to select a date format.

About this task

To select a date format:

Procedure

1. Specify a question mark (?) in the **DATE** field of a previous SELECT Statement Option panel. The Type Values - DATE panel is displayed, as shown in the following figure:

```
INZP006 ----- Type Values ----- Row 45 of 111
Command ==>                               Scroll ==> CSR

Line commands: any character - Select

Input format : ULCTYPE

Select one format and press Enter to validate or PF3 to cancel

S Name          Description          Default
-----
. DATE_A        MM-DD-YYYY (USA with DELIM = '/')
. DATE_B        MM-DD-YY
. DATE_C        YYYY-MM-DD (ISO and JIS)          Y
. DATE_D        YY-MM-DD
. DATE_E        DD-MM-YYYY (EUR with DELIM ='.')
. DATE_F        DD-MM-YY
. DATE_G        YYYY-DDD
. DATE_H        YY-DDD
. DATE_I        MMDDYYYY
. DATE_J        MMDDYY
. DATE_K        YYYYMMDD
. DATE_L        YYMMDD
. DATE_M        DDMMYYYY
. DATE_N        DDMMYY
. DATE_O        YYYYDDD
. DATE_P        YYDDD
. DATE_DB2      DATE in the DB2 default output format
. DATE          DATE in the internal DB2 format in 4 bytes
***** Bottom of data *****
```

Figure 54. Type Values - DATE panel

2. Specify the S selection code to select the date format.

Related tasks:

“Specifying options for the SELECT statement” on page 225

The SELECT Statement Option panels are used to specify options for a SELECT statement.

“Selecting timestamp formats” on page 253

The Type Values - TIMESTAMP panel is used to select a timestamp format.

Selecting time formats

The Type Values - TIME panel is used to select a time format.

Procedure

1. Specify a question mark (?) in the **TIME** field of a previous SELECT Statement Option panel. The Type Values - TIME panel is displayed, as shown in the following figure:

```

INZP006 ----- Type Values ----- Row 62 of 111
Command ==>                               Scroll ==> CSR

Line commands: any character - Select

Input format : ULCTYPE

Select one format and press Enter to validate or PF3 to cancel

S Name          Description          Default
-----
. TIME_A        HH.MM.SS (ISO and EUR, and JIS with DELIM =':')  Y
. TIME_B        HH.MM
. TIME_C        HH.MM AM (USA with DELIM =':')
. TIME_D        HHMMSS
. TIME_E        HHMM
. TIME_DB2      TIME in the DB2 default output format
. TIME          TIME in the internal DB2 format in 3 bytes
***** Bottom of data *****

```

Figure 55. Type Values - TIME panel

2. Specify the S selection code to select the time format.

Related tasks:

“Specifying options for the SELECT statement” on page 225

The SELECT Statement Option panels are used to specify options for a SELECT statement.

“Selecting tables” on page 231

The Table List panel is used to create SELECT statements.

Related reference:

“TIME format types” on page 461

Use the TIME format type to specify the output data format.

Selecting timestamp formats

The Type Values - TIMESTAMP panel is used to select a timestamp format.

Procedure

1. Specify a question mark (?) in the **TIMESTAMP** field of a previous SELECT Statement Option panel. The Type Values - TIMESTAMP panel is displayed, as shown in the following figure:

```

INZP006 ----- Type Values ----- Row 68 of 111
Command ==>                               Scroll ==> CSR

Line commands: any character - Select

Input format : ULCTYPE

Select one format and press Enter to validate or PF3 to cancel

S Name          Description          Default
-----
. DATE_A        MM-DD-YYYY
. DATE_B        MM-DD-YY
. DATE_C        YYYY-MM-DD
. DATE_D        YY-MM-DD
. DATE_E        DD-MM-YYYY
. DATE_F        DD-MM-YY
. DATE_G        YYYY-DDD
. DATE_H        YY-DDD
. DATE_I        MMDDYYYY
. DATE_J        MMDDYY
. DATE_K        YYYYMMDD
. DATE_L        YYMMDD
. DATE_M        DDMMYYYY
. DATE_N        DDMMYY
. DATE_O        YYYYDDD
. DATE_P        YYDDD
. DATE          DATE in the internal DB2 format in 4 bytes
. TIME_A        HH.MM.SS
. TIME_B        HH.MM
. TIME_C        HH.MM AM
. TIME_D        HHMMSS
. TIME_E        HHMM
. TIME          TIME in the internal DB2 format in 3 bytes
. TIMESTAMP_A   YYYY-MM-DD-HH.MM.SS
. TIMESTAMP_B   YYYY-MM-DD-HH.MM.SS.NNNNNN      Y
. TIMESTAMP_C   YYYYMMDDHHMMSS
. TIMESTAMP_D   YYMMDDHHMMSS
. TIMESTAMP_E   YYYYMMDDHHMMSSNNNNNN
. TIMESTAMP_F   YYMMDDHHMMSSNNNNNN
. TIMESTAMP_G   YYYY-MM-DD HH:MM:SS.NNN
. TIMESTAMP     TIMESTAMP in the internal DB2 format in 10 bytes
***** Bottom of data *****

```

Figure 56. Type values - *TIMESTAMP* panel

2. Use the S selection code to select the timestamp format.

Related tasks:

“Specifying options for the SELECT statement” on page 225
 The SELECT Statement Option panels are used to specify options for a SELECT statement.

“Selecting date formats” on page 251
 The Type Values - DATE panel is used to select a date format.

Related reference:

“TIMESTAMP format types” on page 461
 Use the *TIMESTAMP* format type to specify the output data format.

Selecting mask formats

The Type Values - MASK panel is used to select a mask format.

Procedure

Specify a question mark (?) in the **MASK** field of a previous SELECT Statement Option panel. The Type Values - MASK panel is displayed, as shown in the

following figure:

```
INZP006 ----- Type Values ----- Row 99 of 111
Command ==>                               Scroll ==> CSR

Line commands: any character - Select

Input format : ULCTYPE

Select one format and press Enter to validate or PF3 to cancel

S Name          Description          Default
-----
. *.*           Unnecessary 0 will not be added
. *.0           Decimal part will be padded with 0
. *.Z           As *.0 plus the decimal point is always printed
. 0.*           A digit must be present prior the decimal point
. 0.0           The int. part is present and dec. one is padded
. 0.Z           As 0.0 plus the decimal point is always printed
. 00.*          The integer part will be padded with 0
. 00.0          The integer and decimal parts are padded with 0
. 00.Z          As 00.0 plus the decimal point is always printed
***** Bottom of data *****
```

Figure 57. Type values - MASK panel

Chapter 6. DB2 HPU user exit

The DB2 HPU user exit is used to customize the output data set that is created by a SELECT statement.

The exit can be written in any of the following languages:

- Assembler
- C
- COBOL/2

The DB2 HPU exit must reside in an authorized library and is loaded dynamically during DB2 HPU operation. This library must be in either the LINKLIST or in an authorized JOBLIB or STEPLIB.

For exits that are written in COBOL/2 and C, the STEPLIB, JOBLIB, or LINKLIST should also point to the LE/370 run-time libraries.

You do not need to modify DB2 HPU syntax to use the exit. Instead, specify the exit by using the OUTEXIT parameter. If this parameter is not present, DB2 HPU will not call the exit.

The DB2 HPU exit accesses the DB2 row in EXTERNAL format before application of the USER format. The exit can inspect, modify, or discard DB2 rows. However, the exit cannot modify the maximum length of variable-length columns for the unloaded tables.

The following topics provide additional information:

- “Register use”
- “Parameter block”
- “Sample Assembler user exit” on page 259
- “Sample COBOL/2 user exit” on page 261

Related reference:

“SELECT block syntax and description” on page 168

The SELECT statement specifies that a logical unload is to be done and indicates the parameters that are associated with the unload job. The SELECT block is a part of the UNLOAD block.

Register use

DB2 HPU communicates with the USER exit through registers 1 and 15. When the USER exit is called, register 1 contains the address of a parameter block, as described by the EXTPLST DSECT in the sample member INFEXIT. Register 15 contains the return code of the exit.

Parameter block

When the USER exit is called, register 1 contains the address of a parameter block.

The parameter block that is pointed to by register 1 contains the following parameters:

Parameter 1 (offset 0)

A function code (one binary word), which is one of the following values:

- **Function code 0 (process)**

A function code of 0 indicates that all parameters are set, and that the data pointed to by the SQLDA can be checked and modified by the exit. The values of the columns are in EXTERNAL format, that is, they are not in DB2 internal format. For variable-length columns, the length can be changed, but must not be greater than the maximum length (field SQLLEN in the SQLDA).

This function is not called if the initialization function returned a nonzero value.

When the function code is 0, the return codes have the following meanings:

Return code	Description
0	The row pointed to by the SQLDA is to be written to the output data set. A USER format can be used to modify the data after the exit is called.
4	This row is not to be written to the output data set.
8	The utility will stop.

- **Function code 1 (initialization)**

A function code of 1 indicates that the exit was called before the table space was scanned and no data was available. The address of the SQLDA has a 0 value, and parameters 3, 4, and 5 are set. Parameter 6 contains the address of an 840-byte ready-to-use work area, which can be modified by the Exit if needed.

When the function code is 1, the return codes have the following meanings:

Return code	Description
0	The exit must be called (function code 0) for this SELECT.
4	The exit must not be called for this SELECT.
8	The utility must be stopped.

- **Function code 2 (end)**

A function code of 2 indicates that only parameters 1, 5, or 6 should be used. The exit can, for example, deallocate the work area that was allocated during the initialization function.

This function is called only if the initialization function return code was 0.

When the function code is 2, *any value* of the return code is accepted.

Parameter 2 (offset 4)

The address of the SQLDA, which describes the selected columns and points to the data

Parameter 3 (offset 8)

The address of a table name

Parameter 4 (offset 16)

The address of the SSID

Parameter 5 (offset 12)

The user number, as shown in the example in "Sample Assembler user exit" (one binary word)

Parameter 6 (offset 52)

The address of a work area. An 840-byte ready-to-use work area is allocated by the calling function, and the address is provided by this parameter. The Exit function can overwrite this address with the work area address that is allocated during initialization, which is then unallocated at the end of the process.

Parameter 7 (offset 28)

The address of two varchar fields which point to the creator and table name.

Sample Assembler user exit

This sample user exit is provided in member INZEXIT (SINZSAMP library). It is written in Assembler.

```
*****
*
* MEMBER : INZEXIT
*
*****
*
* 5655-AA1
* (C) COPYRIGHT INFOTEL 1996, 2012 ALL RIGHTS RESERVED.
*
*****
*
* THIS MEMBER IS CALLED 3 TIMES DURING THE UNLOADING OF A TABLE.
* R1 CONTAINS THE ADDRESS OF A PARAMETER BLOCK AS DESCRIBED BY
* THE EXTXPST DSECT.
*
* THE MOST IMPORTANT FIELD IS THE POINTER TO THE SQLDA OF THE
* PROCESSED TABLE WHICH GIVES DATAS IN EXTERNAL FORMAT.
*
* FUNCTION 0 :
* PROCESS OF THE DB2 ROW
* RETURN CODES :
* 0 : ROW TO BE WRITTEN IN THE OUTPUT FILE
* 4 : ROW DISCARDED
*
* FUNCTION 1 :
* INITIALIZATION OF THE EXIT
* RETURN CODES :
* 0 : EXIT ACTIVE FOR THIS SELECT STATEMENT
* 4 : DESACTIVATION OF THE EXIT FOR THIS SELECT
*
* FUNCTION 2 :
* TERMINATION OF THE EXIT
* RETURN CODES NOT USED
*
*****
R0 EQU 0
R1 EQU 1
R2 EQU 2
R3 EQU 3
R4 EQU 4
R5 EQU 5
R6 EQU 6
R7 EQU 7
R8 EQU 8
R9 EQU 9
R10 EQU 10
R11 EQU 11
R12 EQU 12
R13 EQU 13
R14 EQU 14
R15 EQU 15
*
* SQLDA DESCRIPTION
*
```

SQLDA	DSECT		
SQLDAID	DS	CL8	ID
SQLDABC	DS	F	BYTE COUNT
SQLN	DS	H	NUMBER OF COLUMNS
SQLD	DS	H	NUMBER OF SELECTED COLUMNS
SQLVAR	DS	0F	BEGIN COLUMNS
SQLDSIZ	EQU	**SQLDA	SIZE OF FIXED PART
*			
SQLVARN	DSECT		COLUMN
SQLTYPE	DS	H	TYPE
SQLLEN	DS	0H	LENGTH
SQLPRCSN	DS	X	DEC PRECISION
SQLSCALE	DS	X	DEC SCALE
SQLDATA	DS	A	ADDRESS OF DATA
SQLIND	DS	A	ADDRESS NULL FIELD INDICATOR
SQLNAME	DS	H,CL30	SIZE AND COLUMN NAME
SQLVSIZ	EQU	**SQLVARN	
*			
EXTXPLST	DSECT		PARAMETERS PASSED TO THE EXIT
EXTXFUNC	DS	F	FUNCTION (0 PROCESS, 1 INIT, 2 TERM)
EXTXASQL	DS	A	ADDRESS OF SQLDA
EXTXATBN	DS	A	ADDRESS OF TABLE IDENTIFICATOR
*			CREATOR(8) / TABLENAME(18)
	DS	H	RESERVED
EXTXNREF	DS	H	REFERENCE NUMBERID
EXTXASSI	DS	A	ADDRESS OF SSID(4)
EXTXAUSR	DS	A	ADDRESS OF USER(8)
EXTXATID	DS	A	ADDRESS OF UTILITY ID(16)
EXTXATB	DS	A	ADDRESS OF TABLE IDENTIFICATOR
			VARCHAR CREATOR/TABLENAME
	DS	5F	RESERVED
EXTXAUWA	DS	A	ADDRESS OF WORKAREA
	DS	4F	FIELDS RESERVED FOR THE EXIT
EXTXAMSG	DS	A	ADDRESSE OF MESSAGE
EXTXLMSG	DS	F	LENGTH OF MESSAGE
EXTXUMSG	DS	CL100	EXIT MESSAGE AREA
EXTXUWA	DS	0F	EXIT WORK AREA
	ORG	EXTXPLST+1024	END OF LIST
*			
GETM	DSECT		
SAVE	DS	18F	LOCAL SAVEAREA
TBCREATL	DS	H	CREATOR LENGTH
TBCREAT	DS	CL8	TABLE CREATOR
TBNAMEL	DS	H	TABLE NAME LENGTH
TBNAME	DS	CL18	TABLE NAME
WORK	DS	CL508	WORKAREA
GETML	EQU	**GETM	LENGTH
*			
INZEXIT	CSECT		
STM	R14,R12,12(R13)		SAVE CALLERS REGISTERS
LR	R12,R15		R12=BASE REGISTER
USING	INZEXIT,R12		ESTABLISH ADDRESSABILITY
LR	R10,R1		GET PARAMETERS
USING	EXTXPLST,R10		"
L	R11,EXTXAUWA		R11=WORKAREA ADDRESS
USING	GETM,R11		WORKAREA ADDRESSABILITY
ST	R13,4(R11)		SAVE CALLERS SAVE AREA ADDRESS
L	R2,EXTXFUNC		GET FUNCTION
SLL	R2,2		* 4 FOR DISPATCHING
B	**4(R2)		
B	PROCESS		FUNCTION 0
B	INIT		FUNCTION 1
B	TERM		FUNCTION 2
*			
INIT	EQU	*	
*			
*			A READY-TO-USE WORKAREA OF 840 BYTES IS ALLOCATED BY THE CALLING
*			FUNCTION. IF MORE SPACE IS NEEDED, A LARGER AREA SHOULD BE
*			ALLOCATED HERE.
*			EXAMPLE:
*	GETMAIN R,LV=GETML		GET A WORKAREA
*	ST R1,EXTXAUWA		SAVE GETMAIN ADDRESS
*	LR R11,R1		
*			
	ST R11,8(R13)		GIVE CALLER MY SAVE AREA ADDRESS
	ST R13,4(R11)		SAVE CALLERS SAVE AREA ADDRESS
	L R3,EXTXATB		ADDRESS OF TABLE IDENTIFICATOR

```

L      R1,0 (R3)      TABLE CREATOR VARCHAR
LH     R2,0( R1)     LENGTH
STH    R2, TBCREATL  STORE LENGTH
BCTR   R2, 0         FOR EXECUTE
EX     R2, MVCOWNER  COPY OWNER

L      R1,4 (R3)      TABLE NAME VARCHAR
LH     R2,0(R1)     LENGTH
STH    R2,TBNAMEL   STORE LENGTH
BCTR   R2,0         FOR EXECUTE
EX     R2,MVCNAME   COPY TABLE NAME
*
B      RC0

MVCOWNER MVC TBCREAT, 2(R1)
MVCNAME  MVC TBNAME, 2(R1)
*
PROCESS  EQU *
L      R3,EXTXASQL   GET SQLDA ADDRESS
USING   SQLDA,R3
LA     R4,SQLDSIZ(,R3) PASS FIXED AREA
USING   SQLVARN,R4
XR     R5,R5         GET NUMBER OF COLUMNS
LH     R5,SQLN
LOOP    EQU *
*
*      PROCESS OF THE TABLE COLUMNS
*
LA     R4,SQLVSIZ(,R4)
BCT    R5,LOOP      NEXT COLUMN
B      RC0          END PROCESS OK
*
TERM    EQU *
*
*      IF A WORKING AREA WAS ALLOCATED, IT SHOULD BE FREED HERE
*      EXAMPLE:
*      L      R1,EXTXAUWA   FREE WORKAREA
*      FREEMAIN R,A=(R1),LV=GETML
*
B      RC0          END TERMINATION OK
*
RC0     EQU *
LA     R15,0
B      RETURN
RC4     EQU *
LA     R15,4
B      RETURN
*
RETURN  EQU *
L      R13,SAVE+4
L      R14,12(R13)
LM     R0,R12,20(R13)
BSM   0,R14
LTOrg
END

```

Sample COBOL/2 user exit

This sample user exit is written in COBOL/2.

```

*-----
IDENTIFICATION DIVISION.
*-----
PROGRAM-ID.    INFEXIT0.
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
*
LINKAGE SECTION.

01  EXITPARMS.
    02  EXIT-FUNCTION-CODE          PIC S9(9) BINARY.
        88  PROCESS-CALL           VALUE ZERO.
        88  INITIALIZE-CALL        VALUE 1.
        88  CLEANUP-CALL           VALUE 2.
    02  EXIT-SQLDA-POINTER         POINTER.

```

```

02 FILLER                                POINTER.
02 FILLER                                PIC XX.
02 EXIT-REFERENCE-NUM                    PIC S9(4) BINARY.
02 EXIT-SSID-POINTER                     POINTER.
02 EXIT-USERID-POINTER                   POINTER.
02 EXIT-UTILID-POINTER                   POINTER.
02 EXIT-TBNAME-POINTER                   POINTER.
02 FILLER                                PIC X(20).
02 EXIT-WORKAREA-POINTER                  POINTER.
02 FILLER                                PIC X(16).
02 EXIT-USERMSG-POINTER                   POINTER.
02 EXIT-MESSAGE-SIZE                      PIC S9(9) BINARY.
02 EXIT-USER-MESSAGE                      PIC X(100).

01 WORKAREA.
02 WORKAREA-TBCREATOR                    PIC X(8).
02 WORKAREA-TBNAME                        PIC X(18).
02 FILLER                                PIC X(818).

01 TABLE-INFO-VARCHAR
02 TABLE-CREATOR-VARCHAR                POINTER.
02 TABLE-NAME-VARCHAR                    POINTER.

01 SSID                                  PIC X(4).
01 USERID                                PIC X(8).
01 UTILID                                  PIC X(16).

01 SQLDA.
02 SQLDAX                                PIC X(8).
02 SQLDABC                                PIC S9(8) COMP.
02 SQLN                                   PIC S9(4) COMP.
02 SQLD                                   PIC S9(4) COMP.
02 SQLVAR OCCURS 1 TO 300 DEPENDING ON SQLN INDEXED BY I.
03 SQLTYPE                                PIC S9(4) COMP.
03 SQLLEN                                  PIC S9(4) COMP.
03 SQLDATA                                POINTER.
03 SQLIND                                  POINTER.
03 SQLINDN REDEFINES SQLIND PIC S9(9) COMP.
03 SQLNAME.
04 SQLNAMEL                                PIC S9(4) COMP.
04 SQLNAMEC                                PIC X(30).

```

```

*-----
PROCEDURE DIVISION USING EXITPARMS.
*-----

```

```

EVALUATE TRUE
  WHEN INITIALIZE-CALL
    PERFORM
      SET ADDRESS OF TABLE-INFO TO EXIT-TBNAME-POINTER
      SET ADDRESS OF WORKAREA TO EXIT-WORKAREA-POINTER
      MOVE 'INITIALISATION OK' TO EXIT-USER-MESSAGE
      MOVE ZERO TO RETURN-CODE
    END-PERFORM
  WHEN PROCESS-CALL
    PERFORM
      SET ADDRESS OF SQLDA TO EXIT-SQLDA-POINTER
      MOVE 'PROCESS OK' TO EXIT-USER-MESSAGE
      MOVE ZERO TO RETURN-CODE
    END-PERFORM
  WHEN CLEANUP-CALL
    PERFORM
      MOVE LOW-VALUES TO WORKAREA
      MOVE 'CLEANUP OK' TO EXIT-USER-MESSAGE
      MOVE ZERO TO RETURN-CODE
    END-PERFORM
  WHEN ANY
    PERFORM

```

```
MOVE 'INVALID FUNCTION CODE' TO EXIT-USER-MESSAGE
MOVE 12 TO RETURN-CODE
END-PERFORM
END-EVALUATE

GOBACK.
```

Chapter 7. Troubleshooting

Use these topics to diagnose and correct problems that you experience with DB2 HPU.

Decreasing the amount of necessary memory

The memory resources that are needed to run an unload significantly depends on the DB2 HPU settings, the specific unload to be processed, and the parallelism degree that is used.

About this task

The amount of memory that is needed to unload data increases along with the degree of parallelism that is used. If an unload job runs out of memory, consider reducing the parallelism degree.

Procedure

For the following conditions that apply to your environment, reduce the parallelism degree by using the specified method:

Multiple table spaces

When multiple table spaces are used in a single UNLOAD statement, such as multiple SELECT statements or the use of LIST, split the unique UNLOAD statement that uses multiple table spaces into several UNLOAD statements that uses fewer table spaces. Splitting the UNLOAD statement will prevent too many tasks from being run in parallel.

You can split the UNLOAD statement without changing it by adding the PARALLELISM keyword to the command by setting VUX030 UTLPARAL *maximum_number*, where *maximum_number* is the maximum number of table spaces that you want to process in parallel.

For example, the following statement ensure that the necessary amount of memory is reduced as much as possible by setting the parallelism for table space processing to 1.

```
TEMPLATE OUTPUT DSN MYFILES.TEST.&TS..S&TMP.  
UNIT WORK SPACE(5,1) CYL  
LISTDEF LIST INCLUDE TABLESPACE MYDB.TS%  
UNLOAD TABLESPACE  
OPTIONS TEMPLATESET( TMP = :SELNUM )  
PARALLELISM ( , , 1)  
DB2 NO  
SELECT *  
FROM LIST(LIST)  
OUTDDN OUTPUT  
FORMAT DSNTIAUL
```

LIST and TEMPLATE

When you use TEMPLATEs to allocate input or output data sets that are needed for to unload tables from a LIST, reduce the necessary memory needed specifying ONDEMAND_RESOURCE_ALLOCATION(YES) for the VUU030/ULOPTNS parameter.

Partitioned table spaces

When partitioned table spaces are involved, you can reduce the number of partitions that are processed in parallel by using several methods.

- Limit the scope of a single UNLOAD statement by using the PART () option. To perform the same unload, you must use multiple UNLOAD statements instead of one. For example, the following example shows a single UNLOAD statement that can be replaced with multiple statements that use the PART () option.

```
UNLOAD TABLESPACE MYDB.PART128
SELECT * FROM ME.MYTABLE
OUTDDN(OUTPUT)
```

To reduce the amount of necessary memory, replace the previous statement with the statements that are shown in the following example:

```
UNLOAD TABLESPACE MYDB.PART128 PART(1:32)
SELECT * FROM ME.MYTABLE
OUTDDN(OUTPUT) ;
UNLOAD TABLESPACE MYDB.PART128 PART(33:64)
SELECT * FROM ME.MYTABLE
OUTDDN(OUTPUT) ;
UNLOAD TABLESPACE MYDB.PART128 PART(65:96)
SELECT * FROM ME.MYTABLE
OUTDDN(OUTPUT) ;
UNLOAD TABLESPACE MYDB.PART128 PART(97:128)
SELECT * FROM ME.MYTABLE
OUTDDN(OUTPUT) ;
```

- Add the MAXPART keyword to the UNLOAD statement or set the ULMAXPAR parameter. In this case, DB2 HPU splits the partitions to be processed into several groups, according to the maximum number of partitions that is specified by MAXPART or ULMAPAR. These groups of partitions are considered to be from different table spaces so they are allowed to be unloaded in parallel. To make sure the parallelism degree is actually reduced, the parallelism between table spaces must be set to 1 either by using the PARALLELISM(...,,1) keyword or by setting the value of the VUX030 UTLPARAL parameter to 1, as shown in the following example.

```
UNLOAD TABLESPACE
MAXPART 16
PARALLELISM( , , 1)
DB2 NO
SELECT *
FROM ME.MYTABLE
OUTDDN (OUTP)
```

Also, consider using PARALLELISM(*maximum_number*,...,...) or VUX025 PARALLEL=*maximum_number* to limit partition parallelism to the value that you specify for *maximum_number*. For example, use the following UNLOAD statement to reduce the amount of memory that is needed to unload a table space with 128 partitions:

```
UNLOAD TABLESPACE
PARALLELISM(32, , 1)
DB2 NO
SELECT *
FROM ME.MYTABLE
OUTDDN (OUTP)
```

| Tools Customizer troubleshooting

| Use this information to diagnose and correct problems that you experience with
| Tools Customizer.

Gathering diagnostic information

Before you report a problem with Tools Customizer to IBM Software Support, you need to gather the appropriate diagnostic information.

Procedure

Provide the following information for all Tools Customizer problems:

- A clear description of the problem and the steps that are required to re-create the problem
- Relevant screen captures
- 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
- The Tools Customizer trace data set
- The Tools Customizer data store data set and the *high_level_qualifier.SCCQTENU* data set

Determining the trace data set name

You will need to identify the name of the trace data set if you cannot allocate the trace data set, the trace data set runs out of space, or IBM Software Support asks for it.

The name of the trace data set depends on the prefix setting in the TSO profile. To identify the name of the trace data set, you must know the prefix setting.

- If PREFIX is set, the name of the trace data set is *prefix.CCQ.TRACE*, where *prefix* is the TSO prefix that you specified in the profile.
- If NOPREFIX is set, the name of the trace data set is *user_ID.CCQ.TRACE*, where *user_ID* is your TSO user ID.

Gathering diagnostic information

Before you report a problem with DB2 HPU to IBM Software Support, you need to gather the appropriate diagnostic information.

Procedure

Provide the following information for all DB2 HPU 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 errors in batch processing, provide the following information:

- The complete job log
- Print output

- The content of the PARMLIB, including all members or at least the INZUTIL member and the INZDSSID member, where *SSID* is the name of the DB2 subsystem or group attach name on which the issue occurs.

Messages and codes

Use the information in these messages to help you diagnose and solve Tools Customizer and DB2 HPU problems.

These topics describe the user abend codes and messages that are issued by DB2 HPU and Tools Customizer.

Topics:

- “Tools Customizer messages”
- “DB2 HPU user abend codes” on page 329
- “DB2 HPU messages” on page 333

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" on page 267. 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" on page 267. 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" on page 267. 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" on page 267. 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" on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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 ISPFTRC command, and press Enter.
3. Issue the Tools Customizer command that issues the error.
4. Specify the ISPFTRC command, and press Enter. The ISPF file tailoring trace data set is created. It adheres the following naming convention: *TSO_ID*.ISPFTRACE, 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" on page 267. 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" on page 267. 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" on page 267. 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" on page 267. 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.

CCQC037W The customization information was not found.

Explanation: To use the JOBLIST command, the customization jobs must be regenerated by using the GENERATEALL command or the G line command.

System action: None.

User response: Issue the GENERATEALL command or the G line command to generate the customization jobs.

CCQC038W The customization information cannot be accessed because the *customization_member* customization member is being used.

Explanation: The customization member that was specified on NNN is currently being used.

System action: None.

User response: Determine why the customization member is in use, release it, and redo the work.

CCQC039I The VERIFY/VERIFYOFF command is not active in Generate mode.

Explanation: The VERIFY/VERIFYOFF command is not active in Generate mode because all values saved in Generate mode must already have been verified.

System action: None.

User response: No action is required.

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" on page 267. 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" on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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.

CCQD124E The customization library *library_name* is in use by another metadata library.

Explanation: A different product or component is using the specified customization library. Specify another metadata library in the Workplace panel.

System action: None.

User response: Specify another customization library qualifier in 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" on page 267. 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" on page 267. 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" on page 267. 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" on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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 store data set *data_set_name*
 | already exists in a different volume.

| **Explanation:** Tools Customizer could not create the
 | specified data set because the specified data set already
 | exists in a different volume. Data store data set names
 | must be unique.

| **System action:** Processing stops.

| **User response:** Specify a different data store data set
 | name.

| **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" on page 267. 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" on page 267. 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 namer 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 *product_name* is not associated with a DB2 entry.

Explanation: The product that you are trying to customize is not associated with a DB2 entry. Before a product can be customized, it must be associated with at least one DB2 entry.

System action: None.

User response: Associate one or more DB2 entries with the product.

CCQD569I The *product_name* product configuration is not associated with a DB2 entry.

Explanation: The configuration for the specified product is not associated with a DB2 entry.

System action: None.

User response: 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.

CCQD583I The *from_DB2* DB2 subsystem was copied to the *to_DB2* subsystem.

System action: None.

User response: No action is required.

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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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" on page 267. 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.

<p>CCQF028E An asterisk was improperly specified in a filter argument.</p> <p>Explanation: An asterisk, which is treated as data, is embedded in the filter arguments. A generic filter argument is specified by placing the asterisk in the last nonblank position of the argument. No rows match the filter arguments, so all rows will be shown.</p> <p>System action: Processing stops.</p> <p>User response: Specify a valid filter argument.</p>	<p>System action: Processing stops.</p> <p>User response: Specify only one or two sort fields.</p>
<p>CCQF029I More DB2 entries are associated with the specified product. All DB2 entries are listed.</p> <p>System action: None.</p> <p>User response: No action is required.</p>	<p>CCQF085E A sort order was specified incorrectly in the <i>sort-command</i> command. A sort order can be specified only after a field name.</p> <p>Explanation: Valid orders are A (for ascending) or D (for descending).</p> <p>System action: Processing stops.</p> <p>User response: Specify a valid sort order after a field name.</p>
<p>CCQF080I The customized jobs for the product that you are customizing are stored in this data set.</p> <p>System action: None.</p> <p>User response: No action is required.</p>	<p>CCQF086E The <i>sort-command</i> command has an invalid sort field. The valid fields are <i>list-of-the-table-column-names</i>.</p> <p>Explanation: An invalid sort field was specified.</p> <p>System action: Processing stops.</p> <p>User response: Specify a valid sort field.</p>
<p>CCQF081I The JCL must be browsed or edited.</p> <p>Explanation: You can either browse or edit the JCL.</p> <p>System action: None.</p> <p>User response: No action is required.</p>	<p>CCQF087E The <i>sort-command</i> command has an invalid sort order. The valid orders are A (for ascending) or D (for descending).</p> <p>Explanation: An invalid sort order was specified.</p> <p>System action: Processing stops.</p> <p>User response: Specify a valid sort order.</p>
<p>CCQF082E The <i>sort-command</i> command has an invalid sort field or order. The valid fields are <i>list-of-column-names</i>. The valid sort orders are A (for ascending) or D (for descending).</p> <p>Explanation: An invalid sort field or order was specified.</p> <p>System action: Processing stops.</p> <p>User response: Specify a valid sort field or order.</p>	<p>CCQF088E No row match the specified filter argument. All rows are shown.</p> <p>Explanation: No rows match the selected values.</p> <p>System action: Processing stops.</p> <p>User response: Specify a matched value for filtering.</p>
<p>CCQF083E The <i>sort-command</i> command is missing a sort field.</p> <p>Explanation: A sort field must be specified.</p> <p>System action: Processing stops.</p> <p>User response: Specify a valid sort field.</p>	<p>CCQF089I Type the search arguments to filter objects. A generic filter argument is a search argument of the form AA*.</p> <p>Explanation: In a generic filter argument, only the characters up to the asterisk (*) are compared. The * must be placed in the last nonblank position of the argument. Asterisks embedded in the argument are treated as data.</p> <p>System action: None.</p> <p>User response: No action is required.</p>
<p>CCQF084E The <i>sort-command</i> command has more than two sort fields specified.</p> <p>Explanation: The specified sort command included more than two sort fields. The sort command can have up to two fields specified.</p>	<p>CCQF110I To show the panel instructions section, specify a slash (/). To hide the panel instructions section, remove the slash.</p> <p>System action: None.</p> <p>User response: No is action required.</p>

CCQF111I To show the Products to Customize section, specify a slash (/). To hide the Products to Customize section, remove the slash. The Product to Customize section can be shown or hidden only on the Customizer Workplace panel.

System action: None.

User response: No is action required.

CCQF112I To show the Usage Notes section, specify a slash (/). To hide the Usage Notes section, remove the slash. The Usage Notes section can be shown only on the Product Parameters, LPAR Parameters, and DB2 Parameters panels.

System action: None.

User response: No action is required.

CCQF113I The specified values have been saved.

System action: None.

User response: No action is required.

CCQF114I Displays the Panel Display Options panel. Use this panel to select which information to display on panels .

System action: None.

User response: No action is required.

CCQF116I The fully qualified name of the data set into which you want to copy the current user profile. If the data set name exceeds 42 characters, enclose the name in quotation marks. ALTER or UPDATE authorization to this data set is required.

System action: None.

User response: No action is required.

CCQF117I The volume name in which the data set will reside. If left blank, the volume name will be decided by the system.

System action: None.

User response: No action is required.

CCQF110I To show the panel instructions section, specify a slash (/). To hide the panel instructions section, remove the slash.

System action: None.

User response: No is action required.

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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. Contact IBM Software Support.

CCQI088I The *command* command is not active in BROWSE mode.

Explanation: The specified command can be entered only in Edit mode.

System action: Processing continues.

User response: Put the panel in Edit mode and reissue the command.

CCQI089I The *command* command is already active.

System action: Processing continues.

User response: No action required.

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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. Contact IBM Software Support.

information” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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.SINZDENU*.

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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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 the 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: 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. Contact IBM Software Support.

CCQI755S The XML structure of the *member_name* parameter metadata member is not valid. When a default value is not specified in the metadata member, the "required" attribute with a value of "true" cannot be specified.

System action: Processing stops.

User response: 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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 in the cannot exceed *maximum_number* characters.

Explanation: The specified attribute contains too many characters.

System action: Processing stops.

User response: See “Gathering diagnostic information” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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" on page 267. 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" on page 267. 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 a 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 error 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. 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” on page 267. Contact IBM Software Support.

| **CCQO078I** Additional configurations were discovered and saved in the data store. All DB2 entries associated with this configuration are listed.

| **System action:** None.

| **User response:** No action is required.

| **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” on page 267. 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.SINZDENU*.

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 The Discover data set *data_set_name* exists on a different volume.

Explanation: The specified Discover data set must exist on the same volume as where it was created.

System action: Processing continues.

User response: Specify a different Discover data set name.

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.

CCQS025I The display options were saved.

System action: None.

User response: No action is required.

CCQS026E The customization library data set *data_set_name* could not be opened because the requester does not have UPDATE authority on this data set.

Explanation: Users must have UPDATE authority to open the customization library data set. Users must have UPDATE authority to open the customization library data set.

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.

CCQS027E The customization library data set *data_set_name* could not be created because the requester does not have ALTER authority on this data set.

Explanation: To create the customization library data set, ALTER authority on the data set must be granted.

System action: Processing stops.

User response: Ensure that ALTER authority for the specific 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.

CCQS029E The customization library data set is not valid. Enter a valid data set name or use the Tools Customizer default: *data_set_name*.

Explanation: The specified data set is invalid.

System action: Processing stops.

User response: Specify a valid data set name.

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 *nm*, where *nm* 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 *nm*, where *nm* is 1 - 99.

CCQS033E A user profile cannot be copied into the same user profile

Explanation: The specified data set cannot be copied into user's own user profile.

System action: Processing stops.

User response: Enter a different data set name.

CCQS034E The shared user profile data set *data_set_name* could not be created because the requester does not have UPDATE authority on this data set or because the data set already exists in another volume serial.

Explanation: To create a shared user profile data set, the requester must have update authority on the data set, and the specified data set name must be unique.

System action: Processing stops.

User response: Ensure that the requester has UPDATE authority on the data set and ensure that the data set name is unique.

CCQS035E The specified data set already has a user profile. Specify a different data set, or press Enter again to replace the existing user profile.

Explanation: Pressing Enter overwrites the previous user profile for the specified data set with user's own user profile.

System action: Processing stops.

User response: Specify a different data set name.

CCQS036E The customization library *data_set_name* already exists in *volume* and cannot be created in a different volume. Enter a different customization library name.

Explanation: The same data set name cannot exist in a different volume.

System action: Processing stops.

User response: Specify a different data set name.

CCQS037E The data set name was either not specified or invalid.

Explanation: The data set name specified does not follow the IBM data set name convention.

System action: Processing stops.

User response: Specify a valid data set name.

CCQS038E The specified data set cannot be used.

Explanation: The specified data sets contain information that supports Tools Customizer, but this data set cannot be used.

System action: Processing stops.

User response: Specify a different data set.

CCQS039E The specified data set has an invalid record format.

Explanation: The specified data set should be saved as a different record format. For example, the record format should be FB (Formatted Block) but it is set to VB (Variable Block).

System action: Processing stops.

User response: Specify a valid record format.

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 HPU.

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 ID has been updated from *edit_from_id* to *edit_to_id*.

System action: Processing continues.

User response: No action is required.

CCQT028I The product configuration ID has been updated from *edit_from_id* to *edit_to_id*, and the description has been updated from *edit_from_des* to *edit_to_des*.

System action: Processing continues.

User response: No action is required.

CCQT029I The product configuration description has been updated from *edit_from_des* to *edit_to_des*.

System action: Processing continues.

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.

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.

DB2 HPU user abend codes

This section contains a list of all of the abends that are issued by DB2 HPU.

1003 DB2 HPU was unable to open a sequential file. Correct the JCL to include the missing DD statement, and resubmit the job. This situation might occur when unloading data from image copies (COPYDDN option) of a partitioned table space stored on tape if some of the image copy data set are stored on the same volume. In such case, system messages requesting the same volume be mounted should be found in the system log, such as in the example below:

```
14.55.19 JOB09615*246 IEF455D MOUNT BRES19 ON 0581 FOR MZLFRDI
UNLOAD OR REPLY
```

```
14.55.45 JOB09615 IEF234K 0581.BRES19,PVT,MZLFRDI,UNLOAD
```

```
14.55.45 JOB09615 INZX081 FR9D216 IMAGE COPY IS BEING READ
FROM DDNAME IC001
```

```
14.55.45 JOB09615 INZX081 FR9D216 IMAGE COPY IS BEING READ
FROM DDNAME IC002
```

```
14.55.45 JOB09615 INZX081 FR9D216 IMAGE COPY IS BEING READ
FROM DDNAME IC003
```

```
14.55.45 JOB09615 INZX081 FR9D216 IMAGE COPY IS BEING READ
FROM DDNAME IC004
```

```
14.55.45 JOB09615 IEF233D M 0581,BRES19,,MZLFRDI;UNLOAD, 939
939
```

```
LABO.F1P.DBTLS00.FRD9D216.D13031.T1612444,
```

```
939 OR RESPOND TO IEF455D MESSAGE
```

```
14.55.45 JOB 09615*247 IEF455D MOUNT BRES19 ON 0581 FOR MZLFRDI
UNLOAD OR REPLY
```

```
14.56.09 JOB 09615 IEA995I SYMPTOM DUMP OUTPUT 942
```

```
942 USER COMPLETION CODE=1003
```

```
942 TIME=14.56.07 SEQ=24286 CPU=0000
```

```
ASID=00D6
```

```
942 PSW AT TIME OF ERROR 078D1000 804BD3EE ILC 2 INTC 0D
```

```
942 ACTIVE LOAD MODULE
```

```
ADDRESS=000070000 OFFSET=004B
```

```
942 NAME=INZUTILK
```

```
942 DATAAT PSW 004BD3E8 - 00181610 0A0D9640 D26A0700
```

```
942 AR/GR 0: 00000000/80000000 1: 00000000/800003EB
```

```
942 2: 00000000/000606D0 3: 00000000/1254A8B0
```

```
942 4: 00000000/12386008 5: 00000000/12540000
```

```
942 6: 00000000/1254A548 7: 00000000/00000001
```

```
942 8: 00000000/1254A930 9: 00000000/12706000
```

```
942 A: 00000000/12541088 B: 00000000/004BCF58
```

942 C: 00000000/004BDF58 D: 00000000/0059F000

942 E: 00000000/804BD3D4 F: 00000002/00000008

If you identify such a situation, disable the parallelism by adding an option such as MAXPART 1 PARALLELISM (,1,).

- 1004** DB2 HPU attempted to allocate or open a VSAM file. The attempt was unsuccessful because an allocation or open failure occurred (see the register 15 return code).
See *DFSMS Macro Instructions for Data Sets for z/OS* for a complete description of the possible return codes and reason codes.
- 1005** An error occurred while accessing a VSAM file (see the register 15 return code).
See *DFSMS Macro Instructions for Data Sets for z/OS* for a complete description of the possible return codes and reason codes.
- 1006** A VSAM CLOSE failed (output). See the output messages.
- 1009** A return code greater than 4 was returned from an MVS sort. Error messages from sort are printed in the UTPRINT data set. Determine the cause of the error. In most cases, error messages from MVS sort identify the problem, and a dump is not needed. If the sort is unable to open the UTPRINT data set, only the messages and the abend are provided. For example, if the UTPRINT DD statement is omitted, add a UTPRINT DD statement in your JCL to obtain detailed error messages for analyzing the error.
- 1010** DB2 HPU detected that an internal subtask did not terminate at the end of processing. A possible cause might be that the DD cards do not correspond to the control statements. See the LOG messages. Contact IBM Software Support, if necessary.
- 1012** DB2 HPU was unable to decompress a row. Check the table space after restore.
- 1013** DB2 HPU detected an error in EDITPROC. Check the libraries and the table space.
- 1015** An attempt was made to perform an unload, but the ddname for the IMAGE COPY file was not in the JCL.
- 1017** A subtask abended, which resulted in a general abend. Examine the error messages that were issued for the original abend (for example, B37 issued by a SORT utility).
- 1019** DB2 HPU detected that one of the WHERE clauses was too complex. Possible causes are that one of the keywords is not supported or is not compatible with SQL syntax.
- 1021** When register R15 is 0, the input image copy contains rows with several versions, and the active table version is 0. An input image copy that has rows with several versions means the table was altered before the image copy was taken and was not reorganized or reloaded in the meantime. When the active table version is 0, it has not been altered since it was created. When register R15 is not 0, an unexpected system page sequence was found.
If register R15 is 0, DB2 HPU cannot use the image copy as input. Select a copy that was run after the table space was reorganized.

If register R15 is 12, this means that DB2 HPU has found an unexpected page number (the page that was read was not the expected one). This control is done if CHECK_INTEGRITY LEVEL1 was requested. In such a case:

- R2 contains the expected page number
- R3 contains the number of the page that has been read

For any other value of R15, ensure the image copy is valid by creating an object and populating it with the image copy. You can use DSN1COPY. If the image copy is valid, contact IBM Software Support.

1022 DB2 HPU detected a problem with the version description (OBDREC) of a record from an input image copy. Register R15 contains the version number of that record. The data from the image copy cannot be unloaded in the following situations:

- The version that is indicated by register R15 is not described in the image copy. The image copy might have been created with the SYSEMPAGES NO option.
- When the image copy was created, the table had been altered *n* times (there are *n* records versions in the image copy), and when the table was unloaded, the table had been altered only *m* times, where $m < n$ (the DB2 catalog only describes *m* records versions).
- When the image copy was created before a REORG that materializes a DROP COLUMN, use the DDLDDN option to provide DB2 HPU with a file that contains the DDL of the table as before the involved DROP COLUMN to unload the data from the image copy.

If none of these situations occurred, contact IBM Software Support and provide the job log for a new run of the unload that failed and the related SYSABEND file.

When the image copy was created before a REORG that materializes a DROP COLUMN, use the DDLDDN option to provide DB2 HPU with a file that contains the DDL of the table as before the involved DROP COLUMN to unload the data from the image copy.

11xx For abend codes that are not described below, DB2 HPU detected an internal error. Contact IBM Software Support.

1129

If reason code 0F is associated with this abend, DB2 HPU detected an unsupported situation due to binary zeroes (i.e. x'00') in char columns to be unloaded. This unusual situation prevents an optimization mechanism to work correctly.

Check if your data may contain binary zeroes. If this is an abnormal content, fix the issue before running the unload job again. If this is a normal content, consider adding FIX(OPTCNV) to the ULOPTNS parameter to the DB2 HPU parmlib.

In any other case (i.e. either the reason code different from 0F or the reason code is 0F but either the above explanation or suggested solution does not apply), this might be an internal error. Contact IBM Software Support.

1134 If reason code 0A is associated with this abend, DB2 HPU detected an error while writing a LOB file into its target library. DB2 HPU received an error notification from the STOW macro (SVC21), and the registers are set to the following specifications:

- R14** The return code of STOW (R15 from STOW)
- R2** The reason code of STOW (R0 from STOW)
- R3** The address of the member name (LOB file) that could not be created
- R4** The address of the library name for which the error occurred

This abend can be issued for the following reasons:

- The target library has run out of space (R14=16 and R2=2871=x'B37', or R2=3383=x'D37').
- Directory space for the target library does not exist (R14=12 and R2=0).
- The member (LOB file) to be created already exists in the target library (R14=4 and R2=0).

See *DFSMS Macro Instructions for Data Sets for z/OS* for a complete description of the possible return codes and reason codes.

If a reason code other than 0A is associated with this abend, DB2 HPU detected an internal error. Contact IBM Software Support.

1137 An error occurred while DB2 Sort was running. The following reason code is issued:

- 2** The requested number of sort operations is incorrect because it exceeds the maximum number of sort operations that can be run in parallel or the requested number is out of range. See register R3 for the requested number of sort operations. If R3 contains a positive, small number, consider decreasing the parallelism degree that is requested by your unload process. If R3 contains an irrelevant value, such as a negative number or a positive number that is too large to be the accurate number of required sort operations, contact IBM Software Support.

If a reason code other than 2 is associated with this abend, DB2 HPU detected an internal error. Contact IBM Software Support.

11xx For abend codes that were not previously described, DB2 HPU detected an internal error. Contact IBM Software Support.

12xx DB2 HPU detected an internal error. Contact IBM Software Support.

20xx DB2 HPU detected an internal error. Contact IBM Software Support.

3001-3005

DB2 HPU detected an internal error. Contact IBM Software Support.

4000 Expected information was not provided by the LISTCAT command. The following WTO is issued prior to the 4000 abend:

INZUT4-n : ERROR DURING LISTCAT FOR *dsname*

where *dsname* is the name of the data set for which the LISTCAT command was issued.

If *n*=1: the data set was located but no data set entry was found (information field length is equal to 0).

If *n*=2: for a CLUSTER component, the AMDSB control block could not be retrieved. The AMDSB control block contains the type of the data set (HURBA or HARBA).

Issue a manual LISTCAT command to determine the error.

40xx DB2 HPU detected an internal error. Contact IBM Software Support.

DB2 HPU messages

This section contains a list of all of the messages that are issued by DB2 HPU.

Each DB2 HPU is accompanied by one of the following return codes:

Return code	Definition
0	Informational. No error occurred.
4	Warning. A condition was detected; you might need to take further action.
8 or 16	Error. An error was detected; processing could not continue.

The return code shown for a message is the condition code from the job step in which the message is issued. If additional messages that have higher condition codes are issued during the same job step, the higher condition code is returned.

The message text uses the short variable name, such as *Vxxxxxxx*, or the short parameter name. Use these short names to find descriptions about them in this user's guide.

INZC002E FATAL ERROR WHILE CHECKING DDNAME *ddname* (RC=*reason code*)

Explanation: A unrecoverable error occurred when checking the ddname that was provided.

User response: Contact IBM Software Support.

Return Code: 8

User response: Allocate a file per partition for image copy files.

Return Code: 8

INZC003E ERROR DDNAME *ddname* IS NOT ALLOCATED AND IS NOT ROOT OF ANY ALLOCATED DDNAMES

Explanation: The specified ddname is not allocated in the execution JCL.

User response: Correct the SYSIN or the DD statements that are allocated in the JCL and resubmit the job.

Return Code: 8

INZC018E UNBALANCED QUOTE

Explanation: An unbalanced quotation mark is found in a file (SYSIN, TEMPLATE, LISTDEF, or DDL). Refer to the previous message to see which file contains the unbalanced quotation mark.

User response: Review the file and correct the quotation mark unbalance.

Return Code: 8

INZC004E INCOMPLETE OBJECT DEFINITION FOR TABLESPACE *dbname.tsname*

Explanation: The definition of the table space is incomplete. The partitioning index is missing.

User response: Create the partitioning index.

Return Code: 8

INZC019E ERROR DDNAME *ddname* NOT ALLOCATED AND NO TEMPLATE FOUND

Explanation: A ddname that was specified in the SYSIN is not allocated in the JCL, is not the root of any allocated ddname in the JCL when it is allowable, and is not defined as a TEMPLATE.

User response: Check the JCL and the SYSIN and correct the problem.

Return Code: 8

INZC005E IMAGE COPIES MUST BE SPECIFIED BY PARTITION, INVALID ALLOCATED DDNAME : *ddname*

Explanation: Image copy files should be specified per partition for a partitioned table space.

INZC020E *stmt1* STATEMENT CANNOT BE MIXED WITHIN *stmt2* STATEMENT

Explanation: Incompatible statements were specified in the SYSIN. For example, UNLOAD PLUS cannot be used within a DB2 HPU UNLOAD statement.

INZC021E • INZGE0004

User response: Check the JCL and the SYSIN and correct the incompatibility.

Return Code: 8

INZC021E LISTDEF *list* CANNOT BE FOUND

Explanation: A LISTDEF that was specified in the SYSIN cannot be found. The LISTDEF definition is searched for first in the SYSIN, and then in the LISTDEF LIBRARY.

User response: Verify that a LISTDEF definition exists in the SYSIN or the LISTDEF LIBRARY.

Return Code: 8

INZC022E LISTDEF NAME *list* IS TOO LONG

Explanation: A LISTDEF name that is longer than 18 characters is specified in the SYSIN.

User response: Change the length of the LISTDEF name to be 18 characters or less.

Return Code: 8

INZCE0322 ERROR : THE MODULE INZUTILB IS NOT APF AUTHORIZED

Explanation: DB2 HPU determined that the INZUTILB module was not APF authorized.

User response: Provide the necessary authorization for the above module, and resubmit the job.

Return Code: 8

INZCE0559 ERROR : UNABLE TO LOAD DSNHDECP FROM THE DSNEXIT LIBRARY

Explanation: DB2 HPU was unable to load the DB2 installation options from the DSNEXIT LIBRARY

User response: Check the value of the VZD007 installation parameter for the specified DB2 subsystem.

Return Code: 8

| **INZDSC01I** The *nnn* value of VZD014 is ignored
| because it is ambiguous for the
| *subsystem-ID* SSID.

| **Explanation:** DB2 HPU V4.1 and earlier releases did
| not require any details for DB2 V9 NFM and later
| releases. For such DB2 subsystems, VZD014 had the
| same value (i.e. 915). In DB2 HPU V4.2 and later
| releases, the exact DB2 level must be specified. This
| message is issued because the Discover process in Tools
| Customizer found the ambiguous 915 value and has
| not generated any value. The related value needs to be
| specified by using Tools Customizer.

| **User response:** No action is required.

| **Return Code:** None.

| **INZDSC02I** The *nnn* value of VZD014 is not
| recognized for the *subsystem-ID* SSID.

| **Explanation:** The value that was discovered for the
| VZD014 is not valid. The value was not stored by the
| Discover process. The related value needs to be
| specified by using Tools Customizer.

| **User response:** No action is required.

| **Return Code:** None.

| **INZDSC03I** The *value-name* value of *Vvariable-ID*
| contains a colon (:).

| **Explanation:** The specified of the indicated variable
| contains a colon, which might cause incorrect
| processing in Tools Customizer. The value is kept as
| specified, and if incorrect processing occurs, Tools
| Customizer will issue the CCQO070W message.

| **User response:** No action is required.

| **Return Code:** None.

INZGE00nn ERROR : WRONG *nn* PARAMETER SPECIFIED IN THE PARMLIB *details*

Explanation: An invalid value was specified in the PARMLIB for the specified parameter. The values for *nn* and *details* are:

nn *details*

00 Refer to the line that follows this message.

01 The variable is missing.

02 The variable must be numeric.

03 The variable must be an 8-digit number.

User response: Modify the parameter value, and resubmit the job.

Return Code: 8

INZGE0004 ERROR: WRONG VALUE SPECIFIED FOR VUM028/DISPLUSR IN THE INZTVAR MEMBER, ITS LENGTH IS GREATER THAN THE MAXIMUM LENGTH FOR THIS PARAMETER

Explanation: DB2 HPU detected a length error in the PARMLIB for the VUM028/DISPLUSR parameter.

User response: Correct the length of the value that was specified for the VUM028/DISPLUSR parameter in the PARMLIB.

Return Code: 8

INZGE0102 INTERNAL ERROR : FILE *file*, LINE *line*. PLEASE CONTACT YOUR TECHNICAL SUPPORT

Explanation: DB2 HPU detected an internal error.

User response: Contact IBM Software Support, and supply the return and reason codes.

Return Code: 8

INZGE0200 ERROR : *ddname* DDCARD SHOULD BE LINE A FILE WITH REFCM=VBS

Explanation: The mentioned *ddname* does not relate to a file with the SPANNED format while this format is requested to process the unload request. DB2 HPU detected an internal error.

User response: Change the allocation parameters for the mentioned *ddname* by specifying the VBS REFCM or consider specifying DFSIGDCB YES to allow DB2 HPU to override the DCB of the provided output file and set its REFCM to VBS.

Return Code: ABENDU1003

INZI006E NO CONVERSION WAS AVAILABLE BETWEEN CCSID *ccsid* AND CCSID *ccsid*

Explanation: A conversion could not be performed because no conversion service was available to perform the specified conversion.

User response: Ensure that the appropriate conversion services are online and that the conversion for the specified *ccsid* is available.

Return Code: 8

INZI007I A CONVERSION WAS REQUESTED BETWEEN CCSID *ccsid* AND CCSID *ccsid*

Explanation: This is an informational message. DB2 HPU requested a conversion between the specified CCSIDs.

User response: No action is required.

Return Code: 0

INZI008E A CHARACTER CONVERSION FAILED BETWEEN CCSID *ccsid source* AND CCSID *ccsid target*

Explanation: A character was found in the source string that cannot be converted into the CCSID target, and the PARMLIB parameter VZM008/SCUNSUB was set to NO.

User response: Set the PARMLIB parameter VZUM028/SCUNSUB to YES to allow substitution character in conversions.

Return Code: 8

INZI009E CUNLCNV ERROR: RC *n*, REASON *n* CONVERSION BETWEEN CCSID *ccsid1* AND CCSID *ccsid2*

Explanation: An error occurred when using Conversion Services to convert a string. The return code and reason code were returned from IBM Conversion Services.

User response: Check the return and reason codes in *z/OS Support for Unicode: Using Conversion Services*.

Return Code: 8

INZI010E INTERNAL CONVERSION ERROR BETWEEN CCSID *ccsid1* AND CCSID *ccsid2*

Explanation: An internal error occurred when using Conversion Services to convert a string.

User response: Contact IBM Software Support, and supply the return and reason codes.

Return Code: 8

INZI020I DB2 SUB SYSTEM *ssid* [*group*] DB2 VERSION *version* DSNEXT *dsname* DECIMAL POINT PERIOD|COMMA SQL STRING DELIMITER DEFAULT|QUOTE|APOST MIXED YES|NO ENCODING SCHEME EBCDIC|ASCII|UNICODE EBCDIC CCSID *nnnn,nnnn,nnnn* ASCII CCSID *nnnn,nnnn,nnnn* UNICODE CCSID *nnnn,nnnn,nnnn* DATE FORMAT ISO|USA|EUR|JIS|LOCAL TIME FORMAT ISO|USA|EUR|JIS|LOCAL DECIMAL ARITHMETIC DEC15|DEC31

Explanation: This message indicates the parameters for the DB2 subsystem. The actual output of this message is shown in the following format:

User response: No action is required.

Reason code: None.

INZI049E THE FOLLOWING USS DIRECTORY DOES NOT EXIST: *directory_name*.

Explanation: A directory must exist before it can be used in a template.

User response: Create the target directory before you reference it in a template.

Return Code: 8

INZI142E ERROR: DDNAME *ddname* NOT ALLOCATED

Explanation: An internal error occurred. DB2 HPU has determined that the specified *ddname* was not allocated.

User response: Contact IBM Software Support, and supply the return and reason codes.

Return Code: 8

INZI143E ERROR: DURING *ddname* ALLOCATION

Explanation: DB2 HPU was unable to allocate a temporary data set under *ddname*. If the *ddname* is SYSIN, an extra blank line might be coded after the end of the in-stream SYSIN data set.

User response: Ensure that the parameters for the VUM018/WRKVOL PARMLIB parameter, the VUA007/WRKTUNIT PARMLIB parameter, or both are valid. Also, ensure that all the volumes that these parameters refer to have not run out of free space. Resubmit the job when the VUM018/WRKVOL PARMLIB parameter, the VUA007/WRKTUNIT PARMLIB parameter, or both refer to a set of existing volumes with enough free space.

If the *ddname* is SYSIN and the in-stream SYSIN is followed by an extra blank line, remove the blank line, and resubmit the job.

If the problem persists, contact IBM Software Support, and supply the return codes and reason codes.

Return Code: 8

INZI144E ERROR: DURING DEALLOCATION OF *ddname* *ddname*

Explanation: DB2 HPU was unable to deallocate the *ddname* *ddname*.

User response: Contact IBM Software Support, and supply the return and reason codes.

Return Code: 8

INZI145E ERROR: ALLOCATING THE DDNAME : *ddname* TO THE DSNAME : *dsname*

Explanation: DB2 HPU was unable to allocate one of the following files in the *ddname* and *dsname* that were specified:

- DB2 DSNEXIT
- UNICODE CONVERSION LIBRARY
- IMAGE COPY (LAST_IC or *n*th most recent copy)

User response: Check the following files for the failed allocation:

- For the DB2 DSNEXIT file, check the content of PARMLIB parameter VZD007.

- For the UNICODE CONVERSION SERVICES file, check the content of PARMLIB parameter VZM006/SCUNMOD.

- For allocation of the image copy, check if the image copy file exists.

Return Code: 8

INZI146E ERROR WHEN SUBMITTING AN SQL QUERY TO DB2 ON PLAN *planname*

Explanation: DB2 cannot reply to an SQL query.

User response: Examine the DB2 messages in the SYSPRINT file. See *DB2 Universal Database™ for z/OS Messages and Codes* for a description of the possible return codes and reason codes.

Return Code: 8

INZI147E ERROR WHEN CLOSING UTILITY PLAN, ERROR CODE: *nn* REASON CODE: *nn*

Explanation: A DB2 CAF closing error occurred.

User response: See the *DB2 for z/OS Application Programming and SQL Guide* for CAF return code and reason code information.

Return Code: 8

INZI148E ERROR WHEN DISCONNECTING FROM DB2, ERROR CODE: *nn* REASON CODE: *nn*

Explanation: DB2 HPU encountered an error while disconnecting from DB2.

User response: See the *DB2 for z/OS Application Programming and SQL Guide* for CAF return code and reason code information.

Return Code: 8

INZI149S DB2 CONNECTION ERROR, SSID: *ssid*, ERROR CODE: *nn* REASON CODE: *nn*

Explanation: ADB2 CAF connection error occurred.

User response: See the *DB2 for z/OS Application Programming and SQL Guide* for CAF return code and reason code information.

Return Code: 12

INZI150S DB2 OPEN PLAN ERROR, PLAN: *planname*, ERROR CODE: *nn* REASON CODE: *nn*

Explanation: A DB2 CAF open error occurred.

User response: See the *DB2 for z/OS Application Programming and SQL Guide* for CAF return code and reason code information.

Return Code: 12

INZI151E ERROR WHEN FINDING *obj* ERROR CODE: *rc* REASON CODE: *rs*

Explanation: DB2 HPU cannot continue because an internal error occurred when trying to find information in the DB2 control blocks. The search object, *obj*, can be one of the following objects:

- SQLID
- SYSOPERATOR ID
- VCAT
- VERSION NUMBER
- DATACLASS
- MINIMUM DIVIDE SCALE
- DEFAULT FOR TS AND IX PRIMARY QUANTITY
- SYSADM ID
- AUTHID
- SECURITY LABEL

User response: Contact IBM Software Support, and supply the return and reason codes.

Return Code: 16

INZI154E UNEXPECTED INTERNAL CONVERSION *reason*

Explanation: An error occurred during the internal conversion.

User response: Contact IBM Software Support, and supply the reason code.

Return Code: 8

INZI157E ERROR : UNABLE TO LOAD DSNHDECP FROM THE DSNEXIT LIBRARY *dsname*

Explanation: The DSNHDECP module load from the *dsname* library failed.

User response: Check the value of the VZD007 installation parameter for the specified DB2 subsystem.

Return Code: 8

INZI158E ERROR WHEN TRYING TO OBTAIN A SHARED LOCK ON TABLE : *name*

Explanation: An error occurred while positioning a shared lock.

User response: Resubmit the job. If the error occurs again, contact IBM Software Support and supply the return and reason codes.

Return Code: 8

INZI159E ERROR WHEN TRYING TO OBTAIN AN EXCLUSIVE LOCK ON TABLE : *name*

Explanation: An error occurred while positioning an exclusive lock.

User response: Resubmit the job. If the error occurs again, contact IBM Software Support and supply the return and reason codes.

Return Code: 8

INZI161I TIMEOUT WAITING STOP STATUS OF THE *objtype name*

Explanation: A timeout occurred while waiting for the result of a previous STOP command. The object type can be a TABLESPACE or INDEXSPACE. The STOP command is reissued.

User response: No action is required.

Return Code: 0

INZI162E OPERATOR ISSUED CANCEL: DB2 CANNOT STOP THE *objtype name*

Explanation: A CANCEL command was issued. The object type can be a TABLESPACE or INDEXSPACE. This message is issued when DB2 response time is slow.

User response: Increase the values of WAITQTY, WAITQTYM, and WAITUNIT, and resubmit the job.

Return Code: 8

INZI163E TIMEOUT: OPERATOR DID NOT REPLY TO THE CONSOLE: EXECUTION ABORTED

Explanation: DB2 HPU ended abnormally because there was no operator response to continue or cancel the job. This message is issued when DB2 response time is slow.

User response: Increase the values of WAITQTY, WAITQTYM, and WAITUNIT, and resubmit the job.

Return Code: 8

INZI164E ERROR WHEN TRYING TO STOP THE *objtype name* DB2 RETURN CODE : *rc* DB2 REASON CODE : *rs*

Explanation: A STOP request was issued for a table space or index space, but the object might be the object of another active utility. The object type can be a TABLESPACE or INDEXSPACE.

User response: See *DB2 Universal Database for z/OS Messages and Codes* for a description of the possible return codes and reason codes.

Return Code: 8

INZI165E ERROR WHEN TRYING TO STOP THE PARTITION(S) OF *objtype name* DB2 RETURN CODE : *rc* DB2 REASON CODE : *rs*

Explanation: A STOP request was issued for a table space or index space that might be the object of another active utility. The object type can be a TABLESPACE or INDEXSPACE.

User response: See *DB2 Universal Database for z/OS Messages and Codes* for a description of the possible return codes and reason codes.

Return Code: 8

INZI166E ERROR WHEN TRYING TO START THE *objtype name* DB2 RETURN CODE : *rc* DB2 REASON CODE : *rs*

Explanation: A START request was issued for a table space or index space that might be the object of another active utility. The object type can be a TABLESPACE or INDEXSPACE.

User response: See *DB2 Universal Database for z/OS Messages and Codes* for a description of the possible return codes and reason codes.

Return Code: 8

INZI167E ERROR WHEN TRYING TO START THE PARTITION(S) OF *objtype name* DB2 RETURN CODE : *rc* DB2 REASON CODE : *rs*

Explanation: A START request was issued for a table space or index space that might be the object of another active utility. The object type can be a TABLESPACE or INDEXSPACE.

User response: See *DB2 Universal Database for z/OS Messages and Codes* for a description of the possible return codes and reason codes.

Return Code: 8

INZI169E ERROR WHEN OPENING SYSIN

Explanation: An internal error occurred and DB2 HPU was unable to open the SYSIN data set.

User response: Contact IBM Software Support, and supply the return and reason codes.

Return Code: 8

INZI170E ERROR WHEN WRITING SYSIN

Explanation: An internal error occurred and DB2 HPU was unable to write to the SYSIN data set.

User response: Contact IBM Software Support, and supply the return and reason codes.

Return Code: 8

INZI171E ERROR WHEN CLOSING SYSIN

Explanation: An internal error occurred and DB2 HPU was unable to close the SYSIN data set.

User response: Contact IBM Software Support, and supply the return and reason codes.

Return Code: 8

INZI172E *name* DSNUTILB FAILED RETURN CODE = *nn*

Explanation: The specified DB2 DSNUTILB utility cannot continue because it encountered an error.

User response: Examine the DB2 messages in the SYSPRINT file and refer to *DB2 Universal Database for z/OS Messages and Codes* for a description of the return codes and reason codes.

Return Code: 8

INZI173E *name* DSNUTILB ABENDED RETURN CODE = *nn*

Explanation: The DB2 DSNUTILB utility has abended.

User response: Examine the DB2 messages in the SYSPRINT file and refer to *DB2 Universal Database for z/OS Messages and Codes* for a description of the possible return codes and reason codes.

Return Code: 8

INZI174E ERROR WHEN TRYING TO DETERMINE ENFM STATUS

Explanation: A failure occurred when determining the DB2 Version 8 subsystem mode. This error can occur because the DISPLAY GROUP command that was used for this purpose failed.

User response: Check for DB2 messages about the DISPLAY GROUP command. If authorization problems are reported by DB2, ensure that the user who performed the DISPLAY GROUP command has the necessary authorization to do so. An authorized user is the user who submitted the job and the user who is specified by the VUM028/DISPLUSR PARMLIB parameter.

Return Code: 8

INZI175I PROCESSING SYSIN AS *scheme*

Explanation: This is an informational message that indicates the encoding scheme that is used to read the SYSIN.

User response: No action is required.

Return Code: 0

INZI180E ERROR WHEN DISPLAYING THE STATUS OF THE *objtype name*

Explanation: An error occurred while attempting to display the status of the table space or index space. This message is usually preceded by message INZI187E, which contains detailed information about the error. The object type can be a TABLESPACE or INDEXSPACE.

User response: Check the value of PARMLIB parameter VUM028/DISPLUSR. See message INZI187E for more information, and see the *DB2 Universal Database for z/OS Messages and Codes* for a description of the possible return codes and reason codes.

Return Code: 8

INZI181E THE PARTITION *n* OF THE *objtype name* IS NOT AVAILABLE : *status*

Explanation: An UNLOAD of a partitioned table space was requested but the specified partition of the object is unavailable. The object type can be a TABLESPACE or INDEXSPACE.

User response: Modify the status of the object, and resubmit the job.

Return Code: 8

INZI182E THE *objtype name* IS NOT AVAILABLE : *status*

Explanation: An UNLOAD was requested but the specified object is unavailable. The object type can be a TABLESPACE or INDEXSPACE.

User response: Modify the status of the object and restart the utility.

Return Code: 8

INZI183E FAILED TO SWITCH TSO USER *userid* TO EXECUTE DISPLAY COMMAND.

Explanation: The switch from the job user ID to the user ID that was specified in the VUM028/DISPLUSR PARMLIB parameter failed.

User response: Check the VUM028/DISPLUSR PARMLIB parameter and verify that a valid user ID was specified.

Return Code: 8

INZI184E FAILED TO SWITCH BACK TSO USER *userid*

Explanation: The switch from the user ID that was specified in the VUM028/DISPLUSR PARMLIB parameter to the initial job user ID failed.

User response: Contact IBM Software Support.

Return Code: 8

INZI187E DB2 COMMAND ERROR, DB2 RETURN CODE: *rc* REASON CODE: *rs* EXPLANATION: *text*

Explanation: A DB2 command failed. When possible, the result of the command is printed in the SYSPRINT file.

User response: See *DB2 Universal Database for z/OS Messages and Codes* for a description of the possible return codes and reason codes. Check the SYSPRINT file for the result of the command.

Return Code: 8

INZI188E UNABLE TO RETRIEVE VSAM INFORMATION FOR THE *objtype name* [PART *n*] LDS : *dsname*

Explanation: A LISTCAT command failed for the underlying LDS of a DB2 object. The object type can be TABLESPACE or INDEXSPACE. The *dsname* of the LDS is in error and, if the object is partitioned, the partition number is contained in the message.

User response: No action is required.

Return Code: 8

INZI189I DISPLAY GROUP DETAIL INFORMATION : RC=*rc* REASON=*rs*

Explanation: This is an informational message. The DISPLAY GROUP DETAIL that was performed to determine the DB2 MODE through DB2 CAF interface returned a warning (RC=4). The return code and reason code from CAF are also contained in this message.

User response: No action is required.

Return Code: 0

INZI190E ERROR WHEN TRYING TO QUIESCE THE *object name*

Explanation: The table space in a table space quiesce request might be the object of another active utility.

User response: See the *DB2 Codes* or *DB2 Messages* guide for a complete description of the possible error messages that are printed in SYSPRINT.

Return Code: 8

INZI191E ERROR WHEN TRYING TO QUIESCE THE PARTITION(S) OF *object name*

Explanation: The table space in a table space quiesce request might be the object of another active utility.

User response: See the *DB2 Codes* or *DB2 Messages* guide for a complete description of the possible error messages that are printed in SYSPRINT.

Return Code: 8

INZI200E SYNTAX ERROR - UNBALANCED DELIMITERS FOR STRING STARTING AT (*line, column*)

Explanation: The syntax contains a string with unbalanced delimiters.

User response: Check the SYSIN at the location that is indicated to find the string that contains the unbalanced delimiters. Correct the syntax and resubmit the job.

Return Code: 8

INZI201E SYNTAX ERROR - MISSING SHIFT-IN IN STRING STARTING AT (*line, column*).

Explanation: A sequence of DBCS characters is not stopped with a SHIFT_IN character.

User response: Check the SYSIN at the location that is indicated to find the string with the missing SHIFT_IN character. Correct the syntax, and resubmit the job.

Return Code: 8

INZI202E SYNTAX ERROR - MISSING SHIFT-IN IN DBCS IDENTIFIER STARTING AT (*line, column*)

Explanation: A DBCS identifier is not followed by a SHIFT_IN character.

User response: Check the SYSIN at the location that is indicated to find the string with the missing SHIFT_IN character. Correct the syntax, and resubmit the job.

Return Code: 8

INZI203E SYNTAX ERROR - UNBALANCED DELIMITERS IN GRAPHIC STRING STARTING AT (*line, column*)

Explanation: The syntax contains a graphic string with unbalanced delimiters.

User response: Check the SYSIN at the location that is indicated to find the string that contains unbalanced delimiters. Correct the syntax, and resubmit the job.

Return Code: 8

INZI204E SYNTAX ERROR - UNBALANCED DELIMITER IN HEXADECIMAL STRING STARTING AT (*line, column*)

Explanation: The syntax contains a hexadecimal string with unbalanced delimiters.

User response: Check the SYSIN at the location indicated to find the string that contains the unbalanced delimiters. Correct the syntax and resubmit the job.

Return Code: 8

INZI320I LISTDEF INFORMATION: LISTDEF NAME *listdef-name*

Explanation: This is an informational message that indicates a LISTDEF name that is used by DB2 HPU.

User response: No action is required.

Return Code: 0

INZI321E LISTDEF ERROR: INVALID PARTLEVEL(*n*) FOR *name*

Explanation: The PARTLEVEL number that was specified in a LISTDEF is greater than the number of partitions within the specified table space.

User response: Correct the PARTLEVEL specification in the indicated LISTDEF.

Return Code: 8

INZI322E LISTDEF ERROR: THE *option_name* OPTION IS NOT SUPPORTED.

Explanation: DB2 HPU does not support the following options when they are specified in the LISTDEF:

- ALL
- XML
- LOB
- BASE
- HISTORY

User response: Remove the unsupported options from the LISTDEF definition.

Return Code: 8

INZI323E TEMPLATE ERROR: DUPLICATE OPTION *option* IN TEMPLATE *template-name*

Explanation: A template option is defined multiple times.

User response: Modify the template definition to remove the duplicate options.

Return Code: 8

**INZI324E LISTDEF ERROR: OPTION
PARTLEVEL CANNOT BE MIXED
WITH RI OPTION**

Explanation: PARTLEVEL and RI are defined in the same LISTDEF.

User response: Modify the LISTDEF definition.

Return Code: 8

**INZI325I TEMPLATE INFORMATION:
TEMPLATE NAME *template-name***

Explanation: This is an informational message that indicates a template name that was used by DB2 HPU.

User response: No action is required.

Return Code: 8

**INZI326E TEMPLATE ERROR: VOLCNT
PARAMETER MUST BE FROM 1 TO 59
FOR DASD DATASET**

Explanation: An invalid VOLUME COUNT is set in a template for a DASD data set.

User response: Change the VOLCNT parameter in the template definition.

Return Code: 8

**INZI327E TEMPLATE ERROR: WHEN
SPECIFYING A DEVICE NUMBER,
UNCNT MUST BE SET TO 1.**

Explanation: An invalid UNIT COUNT is set in a template, when a physical DEVICE NUMBER is used.

User response: Remove the UNCNT parameter from the template or set UNCNT to 1.

Return Code: 8

**INZI328I DATASET ALLOCATED.
TEMPLATE=*template-name*
DDNAME=*ddname* DSN=*dsname***

Explanation: This is an informational message. A data set that was generated from a TEMPLATE is allocated by DB2 HPU.

User response: No action is required.

Return Code: 0

**INZI329I DATASET ALLOCATED FOR
PARTITION *part*. TEMPLATE=*template-
name* DDNAME=*ddname* DSN=*dsname***

Explanation: This is an informational message. A data set that was generated from a TEMPLATE for a specific partition is allocated by DB2 HPU.

User response: No action is required.

Return Code: 0

**INZI330E DYNAMIC ALLOCATION FAILED.
TEMPLATE = *template name* DSN=*dsname*
RC=*rc*, REASON=*rs*. SVC99 MESSAGES
ARE REPORTED IN JESMSGL CARD**

Explanation: A dynamic allocation through SCV 99 failed. The SVC 99 return code and the corresponding reason code are contained in the return and reason codes. Dynamic allocation messages IKJnnnnnI are issued in WTO in the JESMSGL card. This message is followed by message INZI332I, which contains dynamic allocation parameters.

User response: Refer to the allocation messages that were issued in WTO to determine the reason for the allocation error. If the dsname is generated from a template, correct the template, and resubmit the job. Otherwise, contact IBM Software Support and supply them with the content of messages INZI330E, INZI332I, and IKJnnnnnI.

Return Code: 8

**INZI331E FREE OF DDNAME *ddname* FAILED,
RC=*rc*, REASON=*rs*. SVC99 MESSAGES
ARE REPORTED IN JESMSGL CARD**

Explanation: A dynamic free through SCV 99 failed. The SVC 99 return code and corresponding reason code are contained in the return and reason codes. Dynamic allocation messages IKJnnnnnI are issued in WTO in the JESMSGL card. This message is followed by information message INZI332I, which displays dynamic allocation parameters.

User response: Contact IBM Software Support and supply them with the content of messages INZI330E, INZI332I, and IKJnnnnnI.

Return Code: 8

**INZI332I DYNAMIC ALLOCATION
PARAMETER LIST (SVC99): *text***

Explanation: This message displays the dynamic allocation parameters when a dynamic allocation or free failed.

User response: No action is required.

Return Code: 0

INZI333E SPACE ESTIMATION FAILED

Explanation: DB2 HPU failed to estimate the allocation of the file that was generated by the TEMPLATE statement because a LISTCAT command on the underlying LDS file failed.

User response: Check the status of the table space LDS files.

Return Code: 8

INZI336E DYNAMIC ALLOCATION FAILED.
 TEMPLATE=*template_name* GDG
 BASE=*dsn_gdg* IS NOT DEFINED AND
 GDGLIMIT IS SET TO 0

Explanation: DB2 HPU failed to allocate a GDS file from a template. The GDG base is not defined and the GDGLIMIT is set to zero to prevent its definition.

User response: Do one of the following:

- Change the GDGLIMIT setting to a non-zero value in the SYSIN and then resubmit the job.
- Create the GDG using an IDCAMS DEFINE GENERATIONDATAGROUP command and then resubmit the job.

Return Code: 8

INZI337E DYNAMIC ALLOCATION FAILED
 FOR TEMPORARY FILE=*temp_dsn*
 RC=*rc*, REASON=*reason*

Explanation: The dynamic allocation of a temporary file by using DYNALLOC services (SVC 99) failed. RC and REASON contain the dynamic allocation return and reason codes. Dynamic allocation messages IKJnnnnnl are issued in WTO in the JESMSG card.

User response: Refer to the allocation messages that were issued in WTO to determine the reason for the allocation error.

Return Code: 8

INZI338E IDCAMS ERROR RC=*rc*. *sysprint* of the
 IDCAMS execution which failed

Explanation: The IDCAMS program invoked by DB2 HPU has failed. This error message is followed by the SYSPRINT of the IDCAMS which failed. RC contains the IDCAMS return code.

User response: Refer to the content of the IDCAMS to determine the reason of the error.

Return Code: 8

INZI339E ALLOCATION OF BASE GDG=*gdg_dsn*
 FAILED

Explanation: This message occurs when DB2 HPU failed to create a base GDG for a template file. This error message is generally preceded by error message INZU337E or INZU338E.

User response: Refer to the content of message INZU337E or INZU338E and check the template definition.

Return Code: 8

INZI340E MISSING MODELDCB OR DATACLAS
 IN DEFINITION OF
 TEMPLATE=*template_name*

Explanation: This message occurs when a template that corresponds to GDG files is defined without a MODELDCB or DATACLAS parameter and when the parmlib parameter TEMPLDSCB is left blank.

User response: Check the template definition and parmlib parameter TEMPLDSCB, and resubmit the job.

Return Code: 8

INZI341E MODELDCB=*dsname* IN
 TEMPLATE=*template_name* IS NOT
 CATALOGUED

Explanation: This message occurs when a template contains a MODELDCB that does not correspond to a catalogued entry.

User response: Correct the setting of the MODELDCB parameter in the template definition, and resubmit the job.

Return Code: 8

INZI344I LISTDEF *LISTDEF_name*: *object_type*
object_name NOT FOUND

Explanation: An object that was specified by a LISTDEF statement could not be found. *LISTDEF_name* is the name of the list defined the LISTDEF statement, *object_type* is the type of the missing object among DATABASE, TABLESPACE, INDEXSPACE, TABLE, INDEX, and *object_name* is the name of the missing object.

User response: No action is required.

Return Code: 0

INZI345I LISTDEF *LISTDEF_name*: **CLAUSE -**
INCLUDE_or_EXCLUDE_clause -
IDENTIFIES NO OBJECTS

Explanation: A LISTDEF clause returned no object. *LISTDEF_name* is the name of the list defined the LISTDEF statement, and *INCLUDE_or_EXCLUDE_clause* indicates which clause returned no object.

User response: No action is required.

Return Code: 0

INZI348W THE *option_name* **OPTION IN THE**
template_name **TAPE TEMPLATE IS**
IGNORED.

Explanation: DB2 HPU ignored an option in a template. The option name and template name are indicated in the message text.

User response: Remove the specified option to suppress this warning.

Return Code: 4

INZI436E SYNTAX ERROR IN DDLDDN FILE STARTING AT (line1,column1) AND ENDING AT (line2,column2)

Explanation: A syntax error occurred while analyzing the DDL DDN file. DB2 HPU was unable to determine the cause of the error but the error message contains the location where the error occurred.

User response: Verify that valid syntax was used in the indicated area of the DDLDDN file.

Return Code: 8

INZI500I DDL ANALYSING: PROCESSING DDL FILE DDN=ddname

Explanation: This is an informational message. The analysis of the DDL file has started.

User response: No action is required.

Return Code: 0

INZI501I DDL ANALYSING: IN BLOCK STARTING AT POS(line1, column1) ENDING AT POS(line2, column2)

Explanation: This is an informational message that provides the position of the DDL file that is currently being analyzed.

User response: No action is required.

Return Code: 0

INZI502I DDL ANALYSING: ITEM STARTING AT POS(line, column)

Explanation: This is an informational message that provides the position of the DDL file that is currently being analyzed.

User response: No action is required.

Return Code: 0

INZI503E DDL ANALYSING: SYNTAX ERROR AT POS(line, column)

Explanation: A syntax error occurred in a DDL file during analysis.

User response: Correct the syntax error. Contact IBM Software Support if you are unable to determine the cause of the error.

Return Code: 8

INZI510E DDL ANALYSING: DUPLICATE DESCRIPTION OF objtype name

Explanation: An object is defined twice in the DDLDDN file. This message is preceded by informational message INZI502I, which contains the location of the duplication in the DDL file.

User response: Check the CREATE clauses in the DDLDDN file and remove the duplicate definition.

Return Code: 8

INZI511E DDL ANALYSING: DUPLICATE DEFINITION OF PARTITION IN CREATE INDEX STATEMENT

Explanation: The CREATE INDEX clause specifies a clustering index partition more than once.

User response: Check the CREATE INDEX clause in the DDLDDN file and remove the duplicate definition.

Return Code: 8

INZI512E DDL ANALYSING: INVALID INDEX KEY IN CREATE INDEX STATEMENT

Explanation: The CREATE INDEX clause specifies an incorrect key.

User response: Check the CREATE INDEX clauses in the DDLDDN file and remove duplicate definitions.

Return Code: 8

INZI513E DDL ANALYSING: UNDEFINED COLUMN IN THE CREATE INDEX STATEMENT

Explanation: The CREATE INDEX clause specifies an unknown column in the DDLDDN file.

User response: Check the CREATE TABLE and CREATE INDEX clauses in the DDLDDN file for invalid column names.

Return Code: 8

INZI514E DDL ANALYSING: UNDEFINED TABLE IN CREATE INDEX STATEMENT

Explanation: An index is defined on a table that was not previously described in the DDLDDN file.

User response: Check the CREATE TABLE and CREATE INDEX clauses in the DDLDDN file for the undefined index.

Return Code: 8

INZI515E DDL ANALYSING: INVALID BUFFER POOL: *name*

Explanation: An invalid buffer pool is specified in the DDL.

User response: Check the CREATE TABLESPACE or CREATE DATABASE clauses in the DDLDDN file.

Return Code: 8

INZI519W *reason* FOR [TABLESPACE/INDEXSPACE/TABLE] *object* [PART] *part* WAIT(*wait time*)/RETRY(*nb retry*) MECHANISM IS STARTING

Explanation: DB2 HPU detects an unavailable resource. The wait/retry mechanism is activated for one of the following reasons:

- ALLOCATION FAILED
- DB2 STATUS (*status*) NOT SUPPORTED
- LOCK TABLE FAILED

If the resource becomes available, processing continues.

User response: No action is required.

Return Code: None.

INZI520E THE CREATE STATEMENT FOR TABLE *creator.table* IS MISSING THE MANDATORY CLAUSE "IN DATABASE-NAME.TABLESPACE-NAME."

Explanation: The CREATE statement for the specified table does not contain database and table space names, which DB2 HPU requires.

User response: Add the IN DATABASE-NAME.TABLESPACE-NAME clause to the CREATE statement for the specified table.

Return Code: 8

INZI523I UNIT COUNT IS SET TO *unit_count_value*

Explanation: The unit count was set by DB2 HPU for the template file that is indicated in the preceding INZI328I message or the INZI329I message.

User response: No action is required.

Return Code: 0

INZRE40 PAGE NO. *page number* IN LDS NO. *LDS number* OF PAGESET *pageset* IS NOT IN SEQUENCE, SCAN ABORTED

Explanation: Some pages that are not in sequence were read from an image copy file. The INLINE or CHECK keyword was not specified.

User response: Specify INLINE if the input image copy is an INLINE image copy that was created by a LOAD or REORG utility.

Return Code: 8

INZRE41 PAGE NO. *nnnn* IN LDS NO. *nnnn* OF PAGESET *xxxx* IS NOT A VALID DB2 PAGE

Explanation: DB2 HPU encountered a DB2 page that is not valid.

User response: Use the REPAIR command to correct the condition; then resubmit the job. If you are unloading from an image copy file, check that the input file contains a valid image copy.

This message is issued if DB2 HPU encounters an error that is related to the row structure while it is reading the rows of a table space. To limit the number of error messages that are issued, use the VUX018/LDSERRLM variable in the PARMLIB. When the limit that is specified in VUX018/LDSERRLM is reached, message INZRE41 is issued with the following text: INZRE41 MESSAGE LIMIT REACHED, ONE OR MORE MESSAGE(S) NOT PRINTED.

Return Code: 8

INZRE42 ROW AT OFFSET *nnnn* ON PAGE (HEX) *nnnn* OF PAGESET *xxxx* IS INVALID

Explanation: DB2 HPU encountered a row that is invalid.

User response: Use the REPAIR command to correct the condition; then resubmit the job.

This message is issued if DB2 HPU encounters an error that is related to the row structure while it is reading the rows of a table space. To limit the number of error messages that are issued, use the VUX018/LDSERRLM variable in the PARMLIB. When the limit specified in VUX018/LDSERRLM is reached, message INZRE42 is issued with the following text: INZRE42 MESSAGE LIMIT REACHED, ONE OR MORE MESSAGE(S) NOT PRINTED.

Return Code: 8

INZRE45 IMAGE COPY FROM DD *ddname* CONTAINS NO ROW WITH SPECIFIED OBID(S). SCAN ABORTED

Explanation: The image copy data set that is to be unloaded and allocated to the ddname *ddname* contains a table that does not belong to the requested table. The OBID report provides information about the image copy. This report is created only when an image copy of a table space that contains a single table is processed.

User response: Specify the correct OBID value or ORIGINOBID 0.

Return Code: 8

INZR1007 ERROR WHEN OPENING VIRTUAL FILE : DB2FVNNN

Explanation: An internal error occurred when opening virtual file DB2FVNNN.

System action: Processing ends.

User response: Contact IBM Software Support, and supply the return codes and reason codes.

Return Code: 8

INZR1008 ERROR WHEN WRITING VIRTUAL FILE : DB2FVNNN

Explanation: An internal error occurred when writing virtual file DB2FVNNN.

System action: Processing ends.

User response: Contact IBM Software Support, and supply the return and reason codes.

Return Code: 8

INZR2000 DB2 ERROR: PREPARE / DESCRIBE

Explanation: This is a DB2 error. The lines that follow this message contain the error message that is generated by DB2.

System action: Processing ends.

User response: Correct the error.

Return Code: 8

INZR2001 DB2 ERROR: OPEN CURSOR

Explanation: This is a DB2 error. The lines that follow this message contain the error message that is generated by DB2.

System action: Processing ends.

User response: Correct the error.

Return Code: 8

INZR2004 DB2 ERROR: FETCH CURSOR

Explanation: This is a DB2 error. The lines that follow this message contain the error message that is generated by DB2.

System action: Processing ends.

User response: Correct the error.

Return Code: 8

INZR2010 DB2 ERROR: CONNECTING TO *ssid*, RETURN CODE=*rc* REASON CODE=*rs*

Explanation: A DB2 CAF connection error occurred.

User response: See the CAF return codes and reason codes in the *DB2 for z/OS Application Programming and SQL Guide*.

Return Code: 12

INZR2011 DB2 ERROR: OPENING PLAN *planname*, RETURN CODE=*rc* REASON CODE=*rs*

Explanation: A DB2 CAF open error occurred.

User response: See the CAF return codes and reason codes in the *DB2 for z/OS Application Programming and SQL Guide*.

Return Code: 12

INZR2012 DB2 ERROR: DISCONNECTING FROM *ssid*, RETURN CODE=*rc* REASON CODE=*rs*

Explanation: An error occurred while DB2 HPU was disconnecting from DB2.

User response: See the CAF return codes and reason codes in the *DB2 for z/OS Application Programming and SQL Guide*.

Return Code: 8

INZR2013 DB2 ERROR: CLOSING PLAN *planname*, RETURN CODE=*rc* REASON CODE=*rs*

Explanation: A DB2 CAF closing error occurred.

User response: See the CAF return codes and reason codes in the *DB2 for z/OS Application Programming and SQL Guide*.

Return Code: 8

INZT003E TEMPLATE ERROR: INVALID EXPIRATION DATE FORMAT. VALID FORMATS ARE YYYYDDD, YYYY/DDD

Explanation: An invalid expiration date is specified in a template.

User response: Specify a valid date and resubmit the job.

Return Code: 8

INZT004E • INZT014E

INZT004E **TEMPLATE ERROR: INVALID OPERAND** *operand*

Explanation: An invalid operand is specified in a template.

User response: Specify a valid operand and resubmit the job.

Return Code: 8

INZT005E **TEMPLATE ERROR: VOLSER** *volser* **IS TOO LONG**

Explanation: An invalid VOLSER is specified in a template.

User response: Specify a valid VOLSER and resubmit the job.

Return Code: 8

INZT006I **TEMPLATE INFORMATION: TAPE-OPTIONS ARE IGNORED BY UTILITY**

Explanation: DB2 HPU does not support TAPE-OPTIONS of TEMPLATE. These options are ignored.

User response: No action is required.

Return Code: 0

INZT007E **TEMPLATE ERROR: INVALID GDG PARAMETER**

Explanation: The GDG that is specified in a TEMPLATE statement is incorrect.

User response: Correct the incorrect GDG and resubmit the job.

Return Code: 8

INZT008E **TEMPLATE ERROR: OPERAND *n* OF SUBSTRING** *substring* **IS OUT OF RANGE**

Explanation: The parameters specified in a SUBSTRING expression in a template are out of range.

User response: Check the syntax of the DSN in the template, verify that they are in the correct range, and resubmit the job.

Return Code: 8

INZT009E **TEMPLATE ERROR: DUPLICATE TEMPLATE NAME** *template-name*

Explanation: A redefinition of a TEMPLATE occurs in the SYSIN.

User response: Correct the syntax and resubmit the job.

Return Code: 8

INZT010E **TEMPLATE ERROR: TEMPLATEDD** *ddname* **CANNOT BE OPEN**

Explanation: The TEMPLATEDD library of a TEMPLATE cannot be opened.

User response: Verify that the TEMPLATEDD library is spelled correctly in the SYSIN and the JCL.

Return Code: 8

INZT011E **TEMPLATE ERROR: INVALID DSN VARIABLE** *var* **IN TEMPLATE**

Explanation: An invalid variable is used in a dsname definition of a TEMPLATE.

User response: Modify the variable or use the TEMPLATESET option to define it.

Return Code: 8

INZT012E **TEMPLATE ERROR: INVALID SUBSTRING NOTATION** *substring* **IN DSN OPERAND. DSN OPERAND MUST BE ENCLOSED IN SINGLE QUOTATION MARKS**

Explanation: The dsname expression of a TEMPLATE that uses the SUBSTR expression must be enclosed in single quotation marks.

User response: Use single quotation marks around the dsname specification.

Return Code: 8

INZT013E **TEMPLATE ERROR: FATAL SYNTAX ERROR**

Explanation: A syntax error occurred in a TEMPLATE definition. A previous message contains the position in the SYSIN where the analysis failed.

User response: Check the syntax of the TEMPLATE statement, correct any errors, and resubmit the job.

Return Code: 8

INZT014E **TEMPLATE ERROR: DSN OR PATH KEYWORD EXPECTED**

Explanation: A DSN or PATH keyword is missing in a TEMPLATE definition.

User response: Correct the syntax of the TEMPLATE statement, and resubmit the job.

Return Code: 8

INZT015E TEMPLATE ERROR: OPERAND *operand* IS TOO LONG

Explanation: A DSN expression of a TEMPLATE definition contains a qualifier or member that is too long.

User response: Modify the syntax of the DSN definition in the TEMPLATE statement, and resubmit the job.

Return Code: 8

INZT016E TEMPLATE ERROR: INVALID OPERAND *operand* INTEGER IS EXPECTED

Explanation: An invalid operand is used in a TEMPLATE definition. An integer value was expected.

User response: Correct the TEMPLATE statement, and resubmit the job.

Return Code: 8

INZT017E TEMPLATE ERROR: INVALID OPERAND *operand* FOR OPTION *option*

Explanation: An invalid operand is used in one of the options of a TEMPLATE definition.

User response: Correct the TEMPLATE statement, and resubmit the job.

Return Code: 8

INZT018E TEMPLATE ERROR: SYNTAX ERROR FOR OPTION *option*

Explanation: A syntax error occurred in a TEMPLATE definition.

User response: Correct the TEMPLATE statement, and resubmit the job.

Return Code: 8

INZT019E TEMPLATE ERROR: NUMBER OF VOLSER IS GREATER THAN 255

Explanation: The number of VOLSER that were specified in a TEMPLATE definition is greater than the allowable limit of 255.

User response: Correct the TEMPLATE statement, and resubmit the job.

Return Code: 8

INZT020E TEMPLATE ERROR: DSNTYPE PDS REQUIRES DIR OPTIONS

Explanation: A DSNTYPE PDS is specified in a TEMPLATE without a specified directory block option DIR.

User response: Correct the TEMPLATE statement, and resubmit the job.

Return Code: 8

INZT021E TEMPLATE ERROR: OPERAND IS OUT OF RANGE (*val1*, *val2*) FOR OPTION *option*

Explanation: An operand in the TEMPLATE definition is out of range. The message specifies which option is involved and the authorized range (*val1-val2*).

User response: Correct the TEMPLATE statement, and resubmit the job.

Return Code: 8

INZT022E TEMPLATE ERROR: *option1* OPTIONS CANNOT BE MIXED WITH *option2* OPTIONS

Explanation: Incompatible options are used in the TEMPLATE definition.

User response: Correct the TEMPLATE statement, and resubmit the job.

Return Code: 8

INZT023W TEMPLATE WARNING: TEMPLATE *template-name* FOR STORAGE ON TAPE DEVICE IS IGNORED BY UTILITY

Explanation: DB2 HPU does not support the use of templates that are defined with TAPE options. This TEMPLATE definition is ignored and processing continues. If no other TEMPLATE definition exists and if the corresponding DDNAME is not allocated in the JCL, DB2 HPU will fail with error message INZC019E.

User response: If necessary, change the TEMPLATE definition or use another TEMPLATE.

Return Code: 4

INZT024E TEMPLATE ERROR: INVALID DISP SPECIFICATION

Explanation: A disposition option in the TEMPLATE definition is invalid.

User response: Correct the TEMPLATE statement, and resubmit the job.

Return Code: 8

INZT025E TEMPLATE ERROR: INVALID OPERAND(S) FOR OPTION *option*. VALID OPERAND(S) MAY BE *values*

Explanation: An invalid operand is used in a TEMPLATE definition. The message specifies which TEMPLATE option is invalid and lists valid values.

INZT026E • INZT036I

User response: Correct the TEMPLATE statement, and resubmit the job.

Return Code: 8

INZT026E TEMPLATE ERROR: INVALID PARENTHETICAL EXPRESSION

Explanation: An invalid parenthetical expression is used in a TEMPLATE definition.

User response: Correct the TEMPLATE statement, and resubmit the job.

Return Code: 8

INZT027E TEMPLATE ERROR: COMMA IS EXPECTED

Explanation: A comma is expected in a TEMPLATE definition.

User response: Correct the TEMPLATE statement, and resubmit the job.

Return Code: 8

INZT028E TEMPLATE ERROR: RETPD AND EXPDL OPTIONS ARE EXCLUSIVES

Explanation: Options RETPD and EXPDL are exclusives in a TEMPLATE definition.

User response: Correct the TEMPLATE statement, and resubmit the job.

Return Code: 8

INZT029E TEMPLATE ERROR: INVALID DSNAME QUALIFIER *qualifier*

Explanation: An invalid qualifier occurs in the dsname expression of a TEMPLATE definition.

User response: Correct the TEMPLATE statement, and resubmit the job.

Return Code: 8

INZT030E TEMPLATE ERROR: DOT IS EXPECTED

Explanation: A DOT character is expected in the dsname expression of a TEMPLATE definition.

User response: Correct the TEMPLATE statement, and resubmit the job.

Return Code: 8

INZT032E TEMPLATE ERROR: VARIABLE *var* IS NOT SUPPORTED IN CONTEXT OF HPU UTILITY

Explanation: A dsname variable that was used in the

dsname expression of a TEMPLATE definition cannot be used by DB2 HPU.

User response: Refer to the list of supported variables, correct the TEMPLATE statement, and resubmit the job.

Return Code: 8

INZT033E TEMPLATE ERROR: DSNAME GENERATED FROM TEMPLATE *template-name* IS INVALID DSN=*dsname*

Explanation: The dsname that was generated from a TEMPLATE definition is invalid. This message is usually issued when the length of the generated dsname is greater than 44 characters.

User response: Correct the TEMPLATE statement, and resubmit the job.

Return Code: 8

INZT034E TEMPLATE ERROR: TEMPLATE *template-name1* REDEFINES DSNAME *dsname* PREVIOUSLY GENERATED FROM TEMPLATE *template-name2*

Explanation: The dsname that was generated from a TEMPLATE statement redefines a dsname that was previously generated. Each dsname that was generated from a TEMPLATE statement must be unique.

User response: Correct the TEMPLATE statement to ensure each dsname is unique, and resubmit the job.

Return Code: 8

INZT035E TEMPLATE ERROR: DSN VARIABLE *var* HAS NO VALUE

Explanation: A dsname variable in the dsname expression of a TEMPLATE definition cannot be substituted.

User response: Correct the TEMPLATE statement, and resubmit the job.

Return Code: 8

INZT036I TEMPLATE *template* AT POS(*line, column*) IS READ FROM LIBRARY *ddname*

Explanation: This is an informational message. A template is defined in a TEMPLATE library.

User response: No action is required.

Return Code: 0

INZT038E **TEMPLATE ERROR: UNSUPPORTED OPTION** *option*

Explanation: An unsupported option is used in a TEMPLATE definition.

User response: Correct the TEMPLATE statement, and resubmit the job.

Return Code: 8

INZT040E **TEMPLATE ERROR: INCOMPATIBLE OR MISSING DSNTYPE. DSNTYPE HFS IS REQUIRED**

Explanation: A TEMPLATE that corresponds to an HFS file is specified without option DSNTYPE HFS.

User response: Specify option DSNTYPE HFS in the TEMPLATE statement and resubmit the job.

Return Code: 8

INZT041E **TEMPLATE ERROR: DSNAME GENERATED FROM TEMPLATE**
template_name IS TOO LONG DSN=*dsn*

Explanation: The dsname that is generated from a TEMPLATE definition is too long. The message displays the generated dsname.

User response: Correct the TEMPLATE statement and resubmit the job.

Reason Code: 8

INZT043E **TEMPLATE ERROR: THE *option_name* TEMPLATE OPTION WITH THE DSN KEYWORD IS NOT VALID.**

Explanation: The PATH options FILEDATA, PATHOPTS, PATHMODE, and PATHDISP cannot be specified with the DSN keyword in the TEMPLATE statement.

User response: Remove the specified PATH option that is not valid.

Return Code: 8

INZT044E **TEMPLATE ERROR: THE PATH NAME AT POS(*line,col*) IS NOT VALID.**

Explanation: A DSN file name cannot be specified as a path name in the PATH option.

User response: Correct the SYSIN, and rerun the job.

Return Code: 8

INZT045E **TEMPLATE ERROR: THE DSN OPTION AT POS(*line,col*) IS NOT VALID.**

Explanation: A path name that is specified with the DSN option must be enclosed in single quotation marks.

User response: Enclose the path name in single quotation marks, and rerun the job.

Return Code: 8

INZT047I **TEMPLATE INFORMATION: THE *option_name* TEMPLATE OPTION IS IGNORED BECAUSE THE PATH OPTION IS SPECIFIED.**

Explanation: When the PATH option is used to create z/OS USS files, the template option is ignored.

User response: If you want to use the PATH option, remove the template option that is ignored. If you do not want to use the PATH option, remove it.

Return Code: 0

INZT048E **TEMPLATE ERROR: THE *option_name* TEMPLATE OPTION CANNOT BE SPECIFIED WITH THE PATH OPTION.**

Explanation: The specified template option is incompatible when the PATH option is used to allocate a z/OS USS file.

User response: Correct the SYSIN, and rerun the job.

Return Code: 8

INZT101I **LISTDEF *list-name* AT POS(*line,column*) IS READ FROM LIBRARY *ddname***

Explanation: This is an informational message. A LISTDEF is defined in a LISTDEF library.

User response: No action is required.

Return Code: 0

INZT102E **LISTDEF ERROR: NAME *list-name* IS TOO LONG.**

Explanation: A LISTDEF name exceeds the 18 character limit.

User response: Correct the LISTDEF statement, and resubmit the job.

Return Code: 8

INZT103E LISTDEF ERROR: INVALID DB-SPEC KEYWORD AT POS *(line,column)*

Explanation: An invalid DB-SPEC keyword is used to define a LISTDEF.

User response: Correct the LISTDEF statement, and resubmit the job.

Return Code: 8

INZT104E LISTDEF ERROR: PARTITION NUMBER OUT OF RANGE FOR PARTLEVEL OPTION

Explanation: A partition number in the PARTLEVEL option of the LISTDEF definition is out of range.

User response: Correct the LISTDEF statement, and resubmit the job.

Return Code: 8

INZT105E LISTDEF ERROR: INVALID DB2 IDENTIFIER *identifier AT POS(line, column)*

Explanation: An invalid DB2 identifier is used in a LISTDEF definition.

User response: Correct the LISTDEF statement, and resubmit the job.

Return Code: 8

INZT106E LISTDEF ERROR: OPTIONS ALL, BASE, LOB, XML ARE EXCLUSIVES

Explanation: The ALL, BASE, LOB, and XML options are exclusives in a LISTDEF definition.

User response: Correct the LISTDEF statement, and resubmit the job.

Return Code: 8

INZT107E LISTDEF ERROR: DUPLICATE LISTDEF NAME *list-name*

Explanation: A duplicate LISTDEF name is used in the SYSIN.

User response: Check the syntax, correct the error, and resubmit the job.

Return Code: 8

INZT108E LISTDEF ERROR: LISTDEF *list-name AT POS(line, column) IS NOT DEFINED*

Explanation: A LISTDEF that was used in the LISTDEF definition is not defined.

User response: Check the name of the included LISTDEF or the name of LISTDEF library that was used, correct the error, and resubmit the job.

Return Code: 8

INZT109E LISTDEF ERROR: LISTDEFDD *ddname* CANNOT BE OPEN

Explanation: The LISTDEFDD library of LISTDEF cannot be opened.

User response: Check the syntax or the JCL for errors, and resubmit the job.

Return Code: 8

INZT110E LISTDEF ERROR: TABLESPACES OR INDEXSPACES MUST BE SPECIFIED WITH DATABASE KEYWORD

Explanation: The filter DATABASE in the LISTDEF definition requires an INDEXSPACES or TABLESPACES keyword.

User response: Correct the LISTDEF statement, and resubmit the job.

Return Code: 8

INZT111E LISTDEF ERROR: INVALID PATTERN *pattern FOR object-type AT POS (line, column)*

Explanation: An invalid pattern is used in the LISTDEF definition.

User response: Correct the LISTDEF statement, and resubmit the job.

Return Code: 8

INZT112E LISTDEF ERROR: INVALID OPERAND *operand*

Explanation: An invalid operand is used in a LISTDEF definition.

User response: Correct the LISTDEF statement, and resubmit the job.

Return Code: 8

INZT113E LISTDEF ERROR: UNSUPPORTED OPTION *option*

Explanation: An unsupported option is used in the LISTDEF definition.

User response: Correct the LISTDEF statement, and resubmit the job.

Return Code: 8

INZU001E SYSIN CANNOT BE OPENED

Explanation: DB2 HPU was unable to open the SYSIN data set.

User response: Ensure that you have a SYSIN DD statement in your JCL. Otherwise, contact IBM Software Support, and supply the return and reason codes.

Return Code: 16

INZU002E EXECUTION ARGUMENTS MISSING

Explanation: The execution arguments are missing.

User response: Provide the missing parameters, and resubmit the job.

Return Code: 16

INZU003E CANNOT DETERMINE THE SSID.

Explanation: An error was detected while determining the SSID parameter.

User response: Correct the parameters, and resubmit the job.

Return Code: 16

**INZU005E ERROR CONVERTING A CONSTANT
IN AN EXPRESSION**

Explanation: An error occurred while doing operations on a literal in an expression (WHERE clause or other).

User response: Check that the literal is correctly specified. Also, ensure that there are no conversion issues. Check to see whether message INZI006E was issued.

Return Code: 8

**INZU008E THE STRIP CHARACTER OF A STRIP
FUNCTION COULD NOT BE
CONVERTED**

Explanation: DB2 HPU was unable to convert the strip character of a strip function.

User response: Check that the literal is specified correctly. Also check to see if there are any conversion issues (message INZI006E).

Return Code: 8 or 16

**INZU009E PART() HAS BEEN SPECIFIED BUT
THE TABLESPACE IS NOT
PARTITIONED**

Explanation: A PART() statement has been detected, but the table space to unload is not partitioned.

User response: Correct the SYSIN.

Return Code: 8

**INZU010I ITEM STARTING AT *position* ENDING
AT *position***

Explanation: This is an informational message. Indicates the position in the SYSIN of the lexical or grammatical element that is causing the message that follows this message.

User response: See the message that follows this message.

Return Code: 0

**INZU011E PARTITION NUMBER OUT OF
RANGE**

Explanation: The partition number that was specified by the PART keyword is greater than the total number of partitions of the table space.

User response: Correct the SYSIN and resubmit the job.

Return Code: 8

**INZU012E CONFLICT BETWEEN SELECT PART()
AND UNLOAD PART()
SPECIFICATIONS**

Explanation: The PART specification of the SELECT statement conflicts with the one in the UNLOAD block. When both are specified, the SELECT PART specification must be a subset of the UNLOAD PART specification.

User response: Correct the SYSIN.

Return Code: 8

**INZU014E INVALID LEXEME AFTER
KEYWORD(S) *keyword-name* VALID
LEXEME(S) IS(ARE) *lexeme***

Explanation: A lexical element was followed by an invalid value.

User response: Correct the SYSIN and rerun the job.

Return Code: 8

**INZU015E *name* STATEMENT MUST BE
TERMINATED BY A SEMICOLON**

Explanation: The end of the specified statement was reached without the presence of a semicolon.

User response: Check that the statement is terminated with a semicolon and that the syntax of the statement is valid.

Return Code: 8

INZU016E *name* STATEMENT SYNTAX IS
INVALID

Explanation: The specified statement is coded incorrectly.

User response: Correct the statement syntax; for example, a parenthesis or comma might be missing.

Return Code: 8

INZU017E INVALID LEXEME. VALID LEXEME(S)
MAY BE *xxxxx*

Explanation: An invalid or unexpected lexeme was found.

User response: Correct the statement syntax.

Return Code: 8

INZU018E INVALID PARMLIB PARAMETER *name*

Explanation: An error occurred while retrieving the indicated PARMLIB parameter.

User response: Check the PARMLIB.

Return Code: 8

INZU019E TABLESPACE NOT FOUND IN THE
DB2 CATALOG

Explanation: The information for the specified table space was not found.

User response: Check that the table space name, if specified in the SYSIN, is correct. Make sure that the catalog information is quiesced in the catalog (QUIESCECAT YES).

Return Code: 8

INZU020E ERROR WHILE RETRIEVING
TABLESPACE INFO FROM THE
CATALOG (*dbname.tsname*)

Explanation: An error occurred while retrieving the information from the catalog.

User response: Check the installation. Ensure that the proper authorizations are in place. Contact IBM Software Support, and supply the return and reason codes.

Return Code: 8

INZU021E UNSUPPORTED CONVERSION *type*
TO *type*

Explanation: An incorrect conversion type was requested.

User response: See "USER format" on page 64 for inconsistency between input and output column type. If you specified the LIKE clause, ensure that it is

compatible with the input data.

Return Code: 8

INZU022E INVALID LEXEME

Explanation: An invalid or unexpected lexeme was found.

User response: Check the statement syntax.

Return Code: 8

INZU023E INVALID PIC SPECIFICATION.

Explanation: An invalid PIC() specification was found.

User response: Check the statement syntax.

Return Code: 8

INZU024E COLUMN *name* IS NOT PART OF THE
TABLE .

Explanation: A column name was found that does not belong to the selected table.

User response: Check the statement syntax.

Return Code: 8

INZU025E INVALID OPERAND FOR OPERATOR
name.

Explanation: An invalid operand was found for the specified operator.

User response: Check the statement syntax.

Return Code: 8

INZU026E MISSING PARENTHESIS.

Explanation: A missing parenthesis has been detected.

User response: Check the statement syntax.

Return Code: 8

INZU027W CNTLCARDS DB2 NOT SUPPORTED,
WILL USE CNTLCARDS DB2LOAD
INSTEAD

Explanation: CNTLCARDS DB2 is not supported. CNTLCARDS DB2LOAD was used instead.

User response: Only the LOAD SYSIN will be generated. To suppress the message, replace DB2 by DB2LOAD.

Return Code: 4

INZU028W CNTLCARDS OPTIONS NOT SUPPORTED, OPTION IGNORED

Explanation: CNTLCARDS options are not supported. The option is ignored.

User response: The CNTLCARDS option that was specified is not supported; no SYSCNTL is generated.

Return Code: 4

INZU030E OUTPUT DDN NOT FOUND FOR PARTITION *number*

Explanation: Output ddname was not found for the specified partition.

User response: Correct the syntax.

Return Code: 8

INZU031E DUPLICATE KEYWORD OR STATEMENT *name*

Explanation: A duplicate specification for the specified keyword or statement has been found.

User response: Correct the syntax.

Return Code: 8

INZU032E INCOMPATIBLE KEYWORD/ STATEMENT *name AND name*

Explanation: The two specified keywords or statements are incompatible.

User response: Correct the syntax.

Return Code: 8

INZU033E CONSTANT EXPECTED

Explanation: A constant was expected.

User response: Correct the syntax.

Return Code: 8

INZU035E OUT OF RANGE (*nn – mm*)

Explanation: The parameter is out of range.

User response: The specified parameter is out of range. A valid range might be given as part of the message.

Return Code: 8

INZU036E MISSING PARAMETERS FOR *name*

Explanation: One or more parameters are missing.

User response: Correct the syntax.

Return Code: 8

INZU037E IN() CLAUSE MUST CONTAIN CONSTANTS ONLY

Explanation: A variable or an expression has been detected in an IN() clause.

User response: Correct the syntax. Change the expression to avoid having a non-constant value in an IN() clause, or authorize the processing by DB2 (DB2 YES, or DB2 FORCE).

Return Code: 8

INZU038E UNSUPPORTED TYPE CONVERSION FOR COLUMN *name*

Explanation: Unsupported type conversion.

User response: The requested output type is incompatible with the actual input type. Correct the syntax. Check the LIKE clause and the COLUMN description of the underlying table or expressions.

Return Code: 8

INZU039E FATAL SYNTAX ERROR BEFORE LINE *nnnn COL nnnn*

Explanation: A fatal syntax error was detected. The syntax analysis stops and the program ends.

User response: The message contains the end of the last lexeme that was read from the syntax. This is not necessarily the cause for the failure. Check the syntax before that point.

Return Code: 8

INZU040E INVALID ORDER BY: SORT KEY *reason*

Explanation: An ORDER BY clause cannot be processed natively by DB2 HPU for one of the following reasons:

- ***i* IS NOT A COLUMN OF TABLE**

DB2 HPU can only issue a sort on columns in the unloaded table, not on expressions. In this case, *i* is the sequence number of the sort key in the ORDER BY clause.

- **EXCEEDS 4088 BYTES LENGTH**

Because of SORT limitations, the total length of the sort keys cannot exceed 4088 bytes.

User response: Modify the ORDER BY clause or use DB2 YES to process the SELECT statement using DB2 access.

Return Code: 8

INZU041E ERROR DURING ALLOCATION OF FIC FOR *xxxx* DSN: *xxxx*

Explanation: The dynamic allocation of a full image copy for a given table space failed.

User response: Make sure that the indicated file is accessible.

Return Code: 8

INZU042E NO GLOBAL FULL IMAGE COPY FOUND FOR TABLESPACE *dbname.tsname*

Explanation: No valid full image copy was found for the specified table space. If the message INZU417I is also issued, see the related explanation for this message for more details on the origin of the issue.

User response: Ensure that DB2 can find the full image copy or copies that are required to process the unload request that you specified in the SYSIN. If you are using a partitioned table space, only *complete* FICs (no partial copies) are considered. Use the PARTITIONED keyword or the ANYTYPE keyword to accept full image copies per partition as input.

Return Code: 8

INZU043E CONVERSION ERROR BETWEEN CCSID *ccsid* AND CCSID *ccsid*

Explanation: DB2 HPU used or might use a conversion between the specified CCSIDs.

User response: No action is required.

Return Code: 8

INZU044W *reason* FOR TABLESPACE *dbname.tsname* ORDER CLUSTER STATEMENT IS IGNORED FOR SELECT *select-number*

Explanation: An ORDER CLUSTER statement was specified, but is ignored for one the following reasons:

NO CLUSTER INDEX FOUND

No cluster index can be found for the table in the SELECT statement or for at least one of the tables resulting from a LISTDEF used in the SELECT statement.

CLUSTER INDEX WITH RANDOM KEY

The cluster index defined on the selected table has a random key and cannot be used for sorting the unloaded data.

CLUSTER INDEX KEY NOT SUPPORTED

FORMAT INTERNAL is specified, the table space is in basic row format (BRF), and the cluster index that is defined on the table in the SELECT statement has a column located after the first variable columns of the table.

User response: Remove the ORDER CLUSTER from the SELECT statement and resubmit the job.

Return Code: 4

INZU047E INVALID EXECUTION ARGUMENTS

Explanation: The execution arguments are incorrect. This error occurs when the PARM parameter of the EXEC statement does not start with "EP=UTLGLCTL", and the UNLOAD statement specifies FASTUNLOAD.

User response: Correct either the execution arguments or the UNLOAD statement.

Return Code: 16

INZU048E WHERE CLAUSE IS NOT BOOLEAN

Explanation: The WHERE clause must evaluate as Boolean.

User response: Correct the WHERE clause so it evaluates to 1 or 0.

Return Code: 16

INZU049W DATEFMT *date-format* FORMAT IS NOT SUPPORTED

Explanation: The *date-format* that is specified in DATEFMT is not supported. The SSID-default date format will be used.

User response: Change the date format to a compatible format, or do nothing if the DB2 default is acceptable.

Return Code: 4

INZU050W TIMEFMT *time-format* FORMAT IS NOT SUPPORTED

Explanation: The time format that is specified in TIMEFMT is not supported. The SSID-default time format is used.

User response: Change the time format to a compatible format, or do nothing if the DB2 default is acceptable.

Return Code: 4

INZU051E DDN *ddname* TOO LONG

Explanation: The specified *ddname* is too long.

User response: Choose a shorter *ddname*. Most situations require a DDN of six characters or less.

Return Code: 8

INZU052E UNSUPPORTED SYNTAX

Explanation: The syntax is not supported. This message is displayed when you use syntax that is compatible with DB2 HPU syntax, but is not syntax supported by DB2 HPU.

User response: Modify the compatible syntax to a supported syntax, or convert the statement into DB2 HPU syntax.

Return Code: 8

INZU053W TSFMT *timestamp-format* FORMAT IS NOT SUPPORTED

Explanation: The TIMESTAMP format that is specified in TSFMT is not supported. The default TIMESTAMP format of the current DB2 subsystem is used.

User response: Change the TIMESTAMP format to a compatible format, or do nothing if the DB2 default is acceptable.

Return Code: 4

INZU055W OUTPUT-FORMAT LOAD NOT SUPPORTED, DSNTIAUL WILL BE USED INSTEAD

Explanation: FASTUNLOAD LOAD output format is not supported. DSNTIAUL is used instead, and processing continues.

User response: No action is required.

Return Code: 4

INZU056E TABLE *name* NOT FOUND IN TABLESPACE *tsname*

Explanation: The specified table was not found in the table space. A possible explanation is that you specified a SELECT statement with the keyword FROM *table*, and the user's name is not the same as the table's creator.

User response: Specify a fully qualified name for the table, if necessary.

Return Code: 8

INZU057E EMPTY TABLESPACE *name*

Explanation: The specified table space did not contain any tables.

User response: The table space that is being unloaded must contain at least one table. Check your object selection in the SYSIN.

Return Code: 8

INZU058E OUTPUT DDN *ddname* NOT FOUND FOR SELECT *number*

Explanation: Output ddname not found for the specified select.

User response: Correct the syntax.

Return Code: 8

INZU059W LOADDDN: *ddname* NOT FOUND. SKIP LOAD STATEMENT GENERATION FOR SELECT STARTING AT POS(*line,column*)

Explanation: The LOADDDN *ddname* was not allocated. No load JCL will be generated for any statement that uses this LOADDDN.

User response: Correct the JCL, the syntax, or both.

Return Code: 0

INZU060E TOO MANY COLUMNS IN INTO CLAUSE FOR SELECT *number*

Explanation: Too many columns were specified in the INTO clause compared to the sum of columns in the SELECT clause.

User response: Correct the INTO Clause.

Return Code: 8

INZU061E INVALID OR UNSUPPORTED DATA TYPE SPECIFICATION

Explanation: Invalid or unsupported data type specification.

User response: Correct the INTO clause.

Return Code: 8

INZU063I UNSUPPORTED SELECT

Explanation: The SELECT statement is not supported by DB2 HPU. An attempt will be made to process the SELECT through DB2.

User response: No action is required.

Return Code: 4

INZU065E INCONSISTENT PARAMETER LIST

Explanation: A multi-parameters function or predicate has inconsistent parameters type.

User response: Check the syntax. Change the expression to eliminate the conflict. Use explicit conversion if necessary or possible.

Return Code: 8

INZU066E NO TABLESPACE FOUND FOR TABLE
name

Explanation: No table space could be found for the specified table.

User response: Correct the syntax. Check the qualification of the table.

Return Code: 8

INZU067E COLUMN FORMAT SPECIFICATION MISSING

Explanation: A column format specification is expected after the comma.

User response: Correct the syntax.

Return Code: 8

INZU068E INVALID COLUMN FORMAT PARAMETER

Explanation: A column format specification contains an invalid parameter.

User response: Correct the syntax.

Return Code: 8

INZU069E INVALID COLUMN FORMAT IDENTIFIER

Explanation: A column format specification does not describe a proper column name or number.

User response: Correct the syntax.

Return Code: 8

INZU070E *element* MISSING

Explanation: The specified element was expected at this point in the syntax.

User response: Correct the syntax.

Return Code: 8

INZU071E UNUSED FORMAT USER COLUMN SPECIFICATION, COLUMN NAME
col_name

Explanation: A column description was given in a FORMAT USER but could not be associated with any column of the SELECT statement.

User response: Verify the SELECT statement syntax.

Return Code: 8

INZU072E UNUSED FORMAT USER COLUMN SPECIFICATION, COLUMN NUMBER
col_number

Explanation: A column description was given in a FORMAT USER but could not be associated with any column of the SELECT statement.

User response: Verify the SELECT statement syntax.

Return Code: 8

INZU073E LIKE-TABLE FORMAT FAILED:
SQL_order

Explanation: A LIKE formatting clause could not be processed. The SQL statement that was used to extract the LIKE information is given.

User response: Verify that the LIKE table exists.

Return Code: 8

INZU074E INVALID MASK VALUE FOR PIC():
mask_value

Explanation: An invalid value was specified for the mask of a PIC() definition.

User response: Check the PIC() definition.

Return Code: 8

INZU075E UNUSED LIKE COLUMN SPECIFICATION, COLUMN NAME
col_name

Explanation: A column description was given in a LIKE clause, but it could not be associated with any column of the SELECT statement.

User response: Verify the SELECT statement syntax.

Return Code: 8

INZU076E UNUSED LIKE COLUMN SPECIFICATION, COLUMN NUMBER
col_number

Explanation: A column description was given in a LIKE clause but could not be associated with any column of the SELECT statement.

User response: Verify the SELECT statement syntax.

Return Code: 8

INZU077E NO OBJECT TO UNLOAD

Explanation: An object (table or table space) to unload was not found in an UNLOAD statement.

User response: Check and modify the UNLOAD and SELECT statements, and resubmit the job.

Return Code: 8

INZU078E CASE STATEMENT CANNOT CONTAIN ONLY NULL EXPRESSION

Explanation: At least one result-expression of a case statement must be not null.

User response: Correct the CASE statement.

Return Code: 8

INZU079E THE TARGET TABLE *table_name* HAS AN INVALID NUMBER OF COLUMN

Explanation: The LIKE table has a wrong column number and the unloaded table does not match the LIKE table.

User response: The LIKE table must have more or as many columns than the table to be unloaded.

Return Code: 8

INZU080E ERROR : SELECT FROM VIEW '*view_name*' IS NOT SUPPORTED BY HPU

Explanation: DB2 HPU does not support the specified view.

User response: See “SELECT block syntax and description” on page 168 for information about defining views.

Return Code: 8

INZU081E ERROR : CLAUSE INTO IS NOT ALLOWED WITH OUTPUT-FORMAT LOAD OR COMMA-DELIMITED

Explanation: Fast Unload does not support an INTO clause with OUTPUT-FORMAT LOAD and COMMA-DELIMITED.

User response: Remove the INTO clause.

Return Code: 8

INZU082W WARNING : CLAUSE STARTING AT (*line, column*) ENDING AT (*line, column*) IS NOT SUPPORTED BY HPU

Explanation: An option of Fast Unload or UNLOAD PLUS syntax is not supported by HPU.

User response: Remove the conflicting statement.

Return Code: 4

INZU083E FORMAT *format* IS NOT SUPPORTED

Explanation: The specified format is not supported by DB2 HPU.

User response: Choose another format.

Return Code: 8

INZU084W WARNING:LOAD-CONTROL FASTLOAD NOT SUPPORTED, DB2LOAD WILL BE USED INSTEAD

Explanation: FASTUNLOAD LOAD-CONTROL FASTLOAD is not supported; DB2LOAD will be used instead.

User response: No action is required.

Return Code: 4

INZU085W WARNING: FIRST ARGUMENT OF SAMPLE STATEMENT IS IGNORED BY HPU

Explanation: You can use the CA FASTUNLOAD SAMPLE keyword with one or two numeric values. However, DB2 HPU only allows you to use this keyword with one value. This message is issued when a SYSIN uses two values with the SAMPLE keyword.

User response: No action is required.

Return Code: 4

INZU086E ERROR : 'ORDER BY' ON EXPRESSION IS NOT SUPPORTED

Explanation: It is not possible to sort output data according to the result of expression. Only DB2 columns are allowed in the ORDER BY clause.

User response: Either specify a DB2 column in the ORDER BY clause or authorize DB2 processing.

Return Code: 8

INZU087E ERROR : MULTIPLE SELECTS WITH ORDER BY CLAUSE UNLOADING IN THE SAME *ddname* ARE NOT ALLOWED

Explanation: Multiple SELECTs on the same table are specified in the same UNLOAD clause. It is not possible to sort the results of multiple SELECT clauses when unloading into the same *ddname*. Only one single ORDER BY clause is allowed by *ddname*.

User response: Either specify different *ddnames* for each SELECT with the ORDER BY clause or split the unload in multiple unload clauses.

Return Code: 8

INZU088E ERROR : SELECTS ON MULTIPLE TABLES WITH ORDER BY CLAUSE UNLOADING IN THE SAME *ddname* ARE NOT ALLOWED

Explanation: Multiple SELECTs on the same table are specified in the same UNLOAD clause. It is not possible to sort the result of a SELECT on one table

and to get the result of a SELECT on another table by using the same ddname.

User response: Either specify a different ddname for the SELECT with the ORDER BY clause or split the UNLOAD clause into two UNLOAD clauses; one UNLOAD clause contains the SELECT with the ORDER BY clause and the other UNLOAD clause contains the remaining SELECT clauses.

Return Code: 8

INZU089E MULTIPLE SELECTS ON A PARTITIONED TABLESPACE WITH DISTINCT ORDER BY CLAUSE ARE NOT ALLOWED

Explanation: Multiple SELECTs on a partitioned table space are specified in the same UNLOAD. It is not possible to have different types of sorts when partitions are unloaded into the same ddname. In this case, all ORDER BY clauses should be the same.

User response: Either specify a different ddname for each type of ORDER BY clauses or split the UNLOAD clause into several UNLOAD clauses.

Return Code: 8

INZU090E UNRECOVERABLE SYNTAX ERROR STARTING AT (line, column) AND ENDING AT (line, column)

Explanation: An unrecoverable syntax error occurred. The incorrect syntax was found within the area indicated by the message, but the exact reason of the error could not be determined.

User response: Check the syntax in the indicated area.

Return Code: 8

INZU091E OPTIONS WITHOUT INFILE ddname OPTION

Explanation: This error occurs in UNLOAD PLUS syntax when using the OBID option without using INFILE options to specify the IMAGE COPY ddname.

User response: Specify the IMAGE COPY ddname with INFILE options.

Return Code: 8

INZU092W OPTION NULLSTRING IS IGNORED BY THE UTILITY

Explanation: The option NULLSTRING (UNLOAD PLUS syntax) used the OBID option without using INFILE options to specify the IMAGE COPY ddname.

User response: Specify the IMAGE COPY ddname with INFILE options.

Return Code: 8

INZU093E THE STRING STARTING AT (line, column) MUST BE ONE CHARACTER LONG

Explanation: The specified string must be one character in length.

User response: Replace the string with a single character.

Return Code: 8

INZU094E ddname ddname NOT FOUND

Explanation: The output ddname that was specified in the SYSIN was not found in the JCL.

User response: Check the output ddname in the JCL.

Return Code: 8

INZU095E SYNTAX ERROR - keyword KEYWORD ONLY VALID WITH block BLOCK

Explanation: The specified keyword cannot be used out of the block that is named in the message.

User response: Verify the statement syntax.

Return Code: 8

INZU096E REQUIRED KEYWORD, keyword, NOT SPECIFIED, PROCESSING TERMINATED

Explanation: The specified keyword was expected at this point in the syntax.

User response: Verify the statement syntax.

Return Code: 8

INZU097E SYNTAX ERROR - UNRECOGNIZED ELEMENT syntax element FOR block BLOCK

Explanation: A syntax element cannot be recognized in the specified syntax block. It might be a misspelled keyword or an invalid argument for the current syntax context.

User response: Verify the statement syntax.

Return Code: 8

INZU098E SYNTAX ERROR - OPERAND, operand, IS USED OUT OF CONTEXT

Explanation: Syntax elements, which are usually used as arguments, were found out of context. Arguments are associated with a keyword that precedes the operand.

User response: Verify the statement syntax.

Return Code: 8

INZU099E SYNTAX ERROR - EMPTY OPTIONS BLOCK

Explanation: An options block cannot be empty.

User response: Verify the statement syntax.

Return Code: 8

INZU100E SYNTAX ERROR - UNLOAD KEYWORD MUST BE FOLLOWED BY TABLESPACE KEYWORD

Explanation: The UNLOAD keyword must be followed by the TABLESPACE keyword.

User response: Verify the statement syntax.

Return Code: 8

INZU101E DB2 NO SPECIFIED, PROCESSING TERMINATED

Explanation: An invalid or an unsupported SYSIN was detected. The selected option DB2 NO implied termination of process.

User response: Correct the SYSIN, or try DB2 YES in case of unsupported SYSIN.

Return Code: 8

INZU102I DB2 FORCE SPECIFIED WILL PROCESS THROUGH DB2

Explanation: The DB2 FORCE option was specified. The UNLOAD will be processed through DB2.

User response: No action is required.

Return Code: 0

INZU103W DB2 YES SPECIFIED, WILL ATTEMPT DB2 PROCESSING

Explanation: An invalid or an unsupported SYSIN was detected. The UNLOAD will be processed through DB2 because the DB2 YES option was specified.

User response: No action is required.

Return Code: 4

INZU104I WARNING IN SELECT STARTING AT (line, column), keyword AND keyword POINT THE SAME BASE ddname: ddname

Explanation: This warning occurs when two of these directives, UNLDDN, OUTDDN, and LOADDN, share the same base ddname.

User response: No action is required.

Return Code: 0

INZU105I *keyword* OPTION IS IGNORED BY THE UTILITY

Explanation: An unsupported OPTION was encountered for FASTUNLOAD or UNLOAD PLUS syntax.

User response: No action is required.

Return Code: 0

INZU106E THE TABLESPACE TO UNLOAD CANNOT BE DETERMINED

Explanation: DB2 HPU cannot determine which table space to unload.

User response: Specify the name of the table space to be unloaded in the SYSIN.

Return Code: 8

INZU107E UNLDDN NOT SPECIFIED

Explanation: A physical unload without UNLDDN was found.

User response: Specify an UNLDDN clause.

Return Code: 8

INZU109E SELECT STATEMENT STARTING AT (line, col) IS TOO LARGE

Explanation: The size of a SELECT statement is larger than the maximum size that is allowed by DB2.

User response: Reduce the size of the indicated SELECT statement.

Return Code: 8

INZU110E ESCAPE IS NOT ALLOWED WITH MIXED STRING IN LIKE PREDICATE

Explanation: The ESCAPE clause is not allowed in the LIKE predicate for a mixed string.

User response: Correct the SYSIN, and resubmit the job.

Return Code: 8

INZU111E RESULT OF SQL EXPRESSION CONCAT IS TOO LONG

Explanation: The result of the CONCAT SQL expression is too long. The maximum length of the resulting string is 65535.

User response: Modify SQL expression CONCAT.

Return Code: 8

INZU112W STRING IN ITEM STARTING AT (*line, columns*) IS TOO LONG AND IS TRUNCATED.

Explanation: The string that was specified in the SYSIN is too long for the item. The string is truncated to the maximum number of characters and processing continues.

User response: Correct the string length in the SYSIN.

Return Code: 4

INZU120W CCSID TARGET OF COLUMN *col_name* IS O. NO CONVERSION WILL BE DONE

Explanation: The CCSID target of the specified output column has been set to 0 by the operator. The column will be unloaded in the source CCSID.

User response: No action is required.

Return Code: 4

INZU121W CCSID TARGET OF COLUMN *col_name* IS O. COLUMN WILL BE UNLOADED IN EBCDIC

Explanation: The CCSID target of the pointed output column has been set to 0 by the operator. The column will be unloaded in the source CCSID, which is EBCDIC CCSID due to internal conversion.

User response: No action is required.

Return Code: 4

INZU122W *charset* SUBSYSTEM CCSID IS SET TO 0. NO CONVERSION WILL BE DONE FOR COLUMN *col_name*

Explanation: The CCSID target of the selected output column is set to 0 and comes from subsystem CCSIDs. The column will be unloaded in the source CCSID. The message indicates which CCSID subsystem is set to 0.

User response: Check if the requested CCSID conversion can be avoided, modify the SYSIN, and resubmit the job.

Return Code: 4

INZU123W *charset* SUBSYSTEM CCSID IS SET TO 0. COLUMN *col_name* WILL BE UNLOADED IN EBCDIC

Explanation: Check if the requested CCSID conversion can be avoided, modify the SYSIN, and resubmit the job.

System action: Modify the installation parameter to get available the CCSID pointed out.

User response: No action is required.

Return Code: 4

INZU124E ERROR IN SELECT STARTING AT (*line, column*), *keyword* AND *keyword* POINT THE SAME BASE *ddname*: *ddname*

Explanation: Two of these directives, UNLDDN, OUTDDN, and LOADDN, share the same base *ddname*.

User response: Specify a unique *ddname* in the SELECT statement.

Return Code: 8

INZU127W DIRECT ACCESS TO DB2 CATALOG FAILED: SWITCH TO SQL ACCESS

Explanation: This is a warning message that indicates that an error occurred when accessing the DB2 catalog by using the DIRECT ACCESS method. The access method is switched to SQL ACCESS.

User response: Contact IBM Software Support to determine the reason of the failure.

Return Code: 0

INZU128E SYSTEM ERROR: DB2 CATALOG EXTRACTION FAILED

Explanation: The catalog extraction failed.

User response: Contact IBM Software Support to determine the reason of the failure.

Return Code: 8

INZU129E UNLOAD FROM IC: SELECT STARTING AT (%i, %i) IS NOT SUPPORTED PROCESSING TERMINATED

Explanation: An unload from an image copy was performed with a SELECT statement that is not supported by DB2 HPU.

User response: Correct the specified SELECT statement.

Return Code: 12

INZU130I CHECK FAILED FOR THE FIC FOR *name*, DEFAULT VALUE WAS TAKEN

Explanation: Due to the parameter VUU032/ULCHKCPY in the PARMLIB, a check was done on the COPYDDN parameter, and no valid full image copy was found for the specified table space in the SYSIBM.SYSCOPY table. The FIC was considered as a NON INLINE FIC.

User response: Ensure that DB2 can find the full image copy or copies that are required to process the

unload request that you specified in the SYSIN. If working on a partitioned table space, only *complete* FICs are considered (no partial copies).

Return Code: 4

INZU131W SUBSYSTEM CCSID IS NOT DEFINED
COLUMN *name* WILL BE UNLOADED
IN DEFAULT SBCS CCSID

Explanation: The MIXED CCSID target for the column is not defined and comes from subsystem CCSIDs. The column will be unloaded in the SBCS subsystem CCSID. The message specifies which subsystem CCSID is not defined.

System action: None.

User response: Verify that the encoding scheme is correctly specified in the UNLOAD command or in the PARMLIB parameter UNLSCHEM. If not, correct it and resubmit the job.

Return Code: 4

INZU133E LOAD STATEMENT ERROR: INVALID
separator SEPARATOR IN FORMAT
DELIMITED, MAXIMUM ALLOWABLE
VALUE IS *limit*

Explanation: A LOAD SYSIN generation (LOADDDN) is requested with FORMAT DELIMITED and the parmlib parameter VUU054/CTRLLIBM is set to YES. This message is issued because DB2 HPU cannot generate a LOAD SYSIN that is compatible with the DB2 LOAD utility, because the field separator (SEP) or the string delimiter (DELIM) used in the FORMAT DELIMITED is not accepted by the DB2 LOAD utility. The limit value displayed in the message text is in hexadecimal.

User response: Correct the value of the SEP or DELIM parameter in the FORMAT DELIMITED definition and resubmit the job.

Return Code: 8

INZU134E INVALID CORRELATION-NAME
correlation-name

Explanation: This is a syntax error. An invalid correlation-name is specified in the select statement.

User response: Check the select statement.

Return Code: 8

INZU145E ERROR: WRONG CPU NUMBER.
EXECUTION STOPPED

Explanation: A module could not run on a specific CPU.

User response: This is a new or upgraded CPU.

Contact IBM Software Support to obtain a new confidential code.

Return Code: 8

INZU146E ERROR: EXPIRATION DATE
EXCEEDED. EXECUTION STOPPED

Explanation: DB2 HPU's expiration date was reached.

User response: Contact IBM Software Support.

Return Code: 8

INZU147E ERROR : DUPLICATE COLUMN *column*
IN ORDER BY CLAUSE

Explanation: Duplicate column is not authorized in an ORDER BY clause.

User response: Check the syntax and remove the duplicate column.

Return Code: 8

INZU168E ERROR WHEN READING DEFAULT
VALUE IN CATALOG FOR COLUMN
name

Explanation: The default value for the specified column in the DB2 catalog cannot be interpreted.

User response: Check the default value for the specified column in the catalog.

Return Code: 8

INZU170E ERROR: UNABLE TO UNLOAD MORE
THAN 1296 REGISTERED IMAGE
COPIES IN THE SAME EXECUTION

Explanation: The maximum number of registered image copies that DB2 HPU can unload at one time was reached. DB2 HPU is unable to allocate more than 1296 ddnames.

User response: Split the SYSIN into smaller parts so that the limit is not reached.

Return Code: 8

INZU172E ERROR: UNABLE TO QUIESCE
TABLESPACE *dbname.tsname*

Explanation: QUIESCE YES LOCK YES was specified but DB2 HPU was unable to quiesce the table space because of its status. When the PARMLIB parameter VUU028/ULQSCEBH is TRY, quiesce is not taken if the object status does not allow it. This can happen when another utility is running at the same time.

User response: If the table space is already quiesced, rerun the job with QUIESCE NO. Otherwise, try to manually quiesce the table space.

Return Code: 8

INZU176I INFORMATION: QUIESCE IMPOSSIBLE. A STOP/START SEQUENCE IS PERFORMED ON TABLESPACE *ixname.tsname*

Explanation: The QUIESCE is impossible on the object because of its status. In this case, the object is stopped and then restarted in order to flush DB2 buffers. The STOP/START sequence is not done when the PARMLIB parameter value of VUU028/ULQSCEBH is TRY.

User response: No action is required.

Return Code: 0

INZU180I UTPRINT DD CARD IN JCL IS NOT USED WHEN VUX020/SORTCLAS IS SPECIFIED IN THE PARMLIB OR WHEN SORTCLASS IS SPECIFIED IN SYSIN

Explanation: The default UTPRINT DDN is not used when the VUX020/SORTCLAS parameter is specified in the PARMLIB or when SORTCLASS is specified in the SYSIN. If a sort is invoked, a UTPRT m file is dynamically allocated.

User response: No action is required.

Return Code: 0

INZU181I WARNING: PARMLIB VARIABLE QUIESCAT IS SET TO OFF

Explanation: The PARMLIB parameter QUIESCAT OFF overrides the SYSIN variable QUIESCAT YES.

User response: No action is required.

Return Code: 0

INZU182I WARNING: PARMLIB VARIABLE QUIESCAT IS SET TO FORCE

Explanation: The PARMLIB parameter QUIESCAT OFF overrides the SYSIN variable QUIESCAT YES.

User response: No action is required.

Return Code: 0

INZU183E ERROR: UNABLE TO QUIESCE DB2 CATALOG. PROCESSING TERMINATED

Explanation: QUIESCECAT YES was specified and but DB2 HPU was unable to quiesce the DB2 catalog. This can happen when another DB2 utility is using the same UTILID, the current utility is in use, or when the user is not authorized to quiesce the DB2 catalog. Check the previous error message for more information.

User response: No action is required.

Return Code: 8

INZU184E ERROR: COLUMN TYPES *column type* ARE NOT SUPPORTED

Explanation: DB2 HPU cannot unload columns of the indicated type. This message occurs when a SELECT statement includes a column with a specific type and the DB2 NO option was specified. The column type is one of the following:

- LOB or XML (IN STREAM): in-stream unloading of LOB or XML data is only supported in SQL mode.
- TIMESTZ: timestamp with time zone is only supported in SQL mode.

DB2 HPU cannot unload column type BLOB, CLOB and DBCLOB. This message occurs when a SELECT STATEMENT includes a column type LOB and SPANNED YES is not used.

User response: Remove the non supported column from SELECT statement if its selection was not intended or use the DB2 YES or DB2 FORCE option to allow the unloading of this column type. Remove the column type BLOB, CLOB, DBCLOB from the SELECT statement or specify SPANNED YES.

Return Code: 8

Example: Unload the whole content of a table having a LOB column in spanned format:

```
UNLOAD TABLESPACE
SELECT *
FROM me.mytable
OUTDDN (SYSREC)
FORMAT VARIABLE ALL
OPTIONS SPANNED YES
LOADDN LOAD
```

INZU185E ERROR: SELECT STATEMENT STARTING AT (*line, column*) INCLUDES UNSUPPORTED COLUMN TYPE

Explanation: The SELECT statement has one or more unsupported column types. DB2 HPU cannot process the statement in any mode (DB2 NO, DB2 YES, DB2 FORCE).

User response: Remove the unsupported column from the SELECT statement, and resubmit the job.

Return Code: 8

INZU186E ERROR: COLUMNS WITH FIELDPROC ARE NOT SUPPORTED IN ORDER CLAUSE

Explanation: A column that was defined with a FIELDPROC is used in the ORDER clause (ORDER BY or ORDER CLUSTER). DB2 HPU does not support such a statement. DB2 HPU can process an ORDER BY statement through DB2. However, it cannot process an ORDER CLUSTER statement.

User response: Authorize DB2 processing for the

ORDER BY clause or replace ORDER CLUSTER with the ORDER BY clause.

Return Code: 8

INZU187E LOAD STATEMENT ERROR: DUPLICATE DELIMITERS IN FORMAT DELIMITED CHARDEL = X' _hexval_', COLDEL = X' _hexval_', DECPT = X' _hexval_'

Explanation: The LOAD SYSIN GENERATION was requested with DELIMITED output format. You cannot specify the same character for more than one type of delimiter (COLDEL, CHARDEL, and DECPT).

User response: Specify a distinct delimiter in the DELIMITED (SEP, DELIM) format. SEP and DELIM delimiters must be different than the decimal separator.

Return Code: 8

INZU188W LOAD STATEMENT WARNING: NO STRING DELIMITERS SPECIFIED

Explanation: LOAD SYSIN GENERATION was requested with DELIMITED output format. No string delimiters are specified, which causes an error in the LOAD UTILITY. For example, when the string contains delimiter characters.

User response: Specify a string delimiter (option DELIM).

Return Code: 4

INZU189E LOAD STATEMENT ERROR: INVALID VALUE X' _hexval_' FOR _delimiter_type DELIMITER IN FORMAT DELIMITED

Explanation: LOAD SYSIN GENERATION was requested with DELIMITED output format. An invalid value was specified for a LOAD UTILITY delimiter (CHARDEL, COLDEL, or DECPT). This message indicates the incorrect value and type of delimiter used.

User response: Change the incorrect value to one that is supported by the LOAD UTILITY.

Return Code: 8

INZU190I IN SELECT STARTING AT POS (line, column), nnnn CONVERSION(S) NEEDED DUE TO EXPRESSION IN COLUMN NUMBER number, FROM CCSID ccsid TO CCSID ccsid

Explanation: This message is issued when a conversion is requested for a column due to an expression. Such a conversion could decrease unload performance. The column is identified by its number in the SELECT clause. The message indicates how many conversions are requested for the column due to an expression.

User response: No action is required.

Return Code: 0

INZU191I IN SELECT STARTING AT POS (line, column), A CONVERSION NEEDED FOR COLUMN col_name, FROM CCSID ccsid TO CCSID ccsid

Explanation: This message is issued when a conversion is requested for a column. The purpose of this message is to inform you that such a conversion could decrease unload performance. The column is identified by its name in the SELECT clause.

User response: No action is required.

Return Code: 0

INZU192I IN SELECT STARTING AT POS (line, column), A CONVERSION NEEDED FOR COLUMN NUMBER number, FROM CCSID ccsid TO CCSID ccsid

Explanation: This message is issued when a conversion is requested for a column. The purpose of this message is to inform you that such a conversion could decrease unload performance. The column is identified by its number in the SELECT clause.

User response: No action is required.

Return Code: 0

INZU193I IN WHERE CLAUSE OF SELECT STARTING AT POS (line, column), nnn CONVERSION(S) NEEDED FROM CCSID ccsid TO CCSID ccsid

Explanation: This message is issued when a conversion is requested in a WHERE clause. The purpose of this message is to inform you that such a conversion could decrease unload performance. The message tells you how many conversions are requested due to the WHERE clause.

User response: No action is required.

Return Code: 0

INZU194W FOR SELECT STARTING AT POS (line, column), SYSTEM EBCDIC WILL BE TAKEN AS DEFAULT SCHEME

Explanation: A SELECT was processed by DB2 and no default CCSID can be determined. Default CCSIDs are determined by the scheme that was specified in the PARMLIB or in the SYSIN. When the scheme is ASIS, default CCSIDs are set to the ones of the table space of the unload task. When SELECT statements are processed by DB2, DB2 HPU sets the default CCSID to the one of the EBCDIC system scheme. These CCSIDs are used to write items such as NULL and NOT NULL indicator values, separator character and string

INZU195I • INZU206E

delimiter for DELIMITED FORMAT and output record padding.

User response: Specify the correct scheme in the SYSIN according to the behavior you require for default CCSIDs.

Return Code: 4

INZU195I INFORMATION : *detail* RC FORCED TO *rc* DUE TO USER PREFERENCES (SEE THE PARMLIB PARAMETER *variable*)

Explanation: This message is issued when the return code is not null (no error). This message informs you that the value of the return code is the value that was specified in the settings. *Detail* is NO ROW UNLOADED and *variable* is VUU24/UNLZLRC, if *rc* indicates that no rows were unloaded.

Details is SWITCH TO SQL MODE and *variable* is VUU069/SQLSWIRC, if *rc* indicates that at least one non supported select has been processed in SQL mode. In all other cases, *detail* is not given.

User response: No action is required.

Return Code: 0

INZU200E LOAD STATEMENT ERROR : DATE FORMAT INVALID FOR LOAD COLUMN *name*

Explanation: This message is issued when LOAD SYSIN GENERATION is requested. This message informs you that the PARMLIB parameter QUIESCAT FORCE overrides the SYSIN variable QUIESCAT YES.

User response: No action is required.

Return Code: 8

INZU201E LOAD STATEMENT ERROR : TIME FORMAT INVALID FOR LOAD COLUMN *name*

Explanation: This message is issued when LOAD SYSIN GENERATION is requested. Format of TIME type column is not compatible with the format that is allowed in LOAD DB2.

User response: Choose an appropriate TIME format for the UNLOAD SYSIN if you want to reload your data.

Return Code: 8

INZU202E LOAD STATEMENT ERROR : TIMESTAMP FORMAT INVALID FOR LOAD COLUMN *name*

Explanation: This message is issued when LOAD SYSIN GENERATION is requested. The format of TIMESTAMP type column is not compatible with the

format that is allowed in LOAD DB2.

User response: Choose an appropriate TIMESTAMP format for the UNLOAD SYSIN if you want to reload your data.

Return Code: 8

INZU203E LOAD STATEMENT ERROR : COLUMN VARIABLE WITHOUT HEADER COLUMN *name*

Explanation: This message is issued when LOAD SYSIN GENERATION is requested. It is not possible to load a variable column without a length header.

User response: Choose an appropriate format for the UNLOAD SYSIN if you want to reload your data.

Return Code: 8

INZU204W LOAD STATEMENT WARNING : COLUMN *col_name* WITH CCSID *ccsid_col* DIFFERENT FROM LOAD CCSID *ccsid_load*

Explanation: This message is issued when LOAD SYSIN GENERATION is requested. The CCSID of the unloaded column is different from the CCSID that was used to load data.

User response: Use a single CCSID for a same UNLOAD file.

Return Code: 4

INZU205W LOAD STATEMENT WARNING : STRING DATA ARE PADDED COLUMN *name*

Explanation: This message is issued when LOAD SYSIN GENERATION is requested. Unloaded data is padded, so data might be different from its original values.

User response: To prevent data modification, do not use the option PADDING.

Return Code: 4

INZU206E LOAD STATEMENT ERROR : REAL LENGTH COLUMN WITH NULL INDICATOR AT END ARE NOT SUPPORTED

Explanation: This message is issued when LOAD SYSIN GENERATION is requested. DB2 HPU cannot load a real length column with a null indicator at the end.

User response: Specify a NULL indicator at the beginning to reload such a column.

Return Code: 12

**INZU209E LOAD STATEMENT ERROR :
INCOMPATIBLE NULL POSITION
FOR COLUMN *name***

Explanation: This message is issued when LOAD SYSIN GENERATION is requested. DB2 HPU cannot load a nullable column with a null ID at the end with the variable format (if the file contains variable columns).

User response: Specify the option NULLPOS BEFORE for such a format.

Return Code: 12

**INZU211W TABLE NAME IN LOAD STATEMENT
load_ddname CANNOT BE
DETERMINED AND NEEDS TO BE
MODIFIED**

Explanation: This message is issued when LOAD SYSIN GENERATION is requested. It is not possible to determine the table name for the LOAD statement. The table name is substituted by xxxxxxxx.xxxxxx card.

User response: Modify the LOAD statement to specify the table name to LOAD.

Return Code: 4

**INZU212W COLUMN *col_name* IN LOAD
STATEMENT IS A CONSTANT. LOAD
STATEMENT MUST BE MODIFY TO
BE USABLE**

Explanation: This message is issued when LOAD SYSIN GENERATION is requested. The column to be loaded is a constant column and does not match a column from the table to load. The LOAD statement is not usable as it is specified.

User response: Modify the LOAD statement.

Return Code: 4

**INZU213W COLUMN *col_name* IN LOAD
STATEMENT RESULT FROM AN SQL
EXPRESSION. LOAD STATEMENT
MUST BE MODIFY TO BE USABLE**

Explanation: This message is issued when LOAD SYSIN GENERATION is requested. The column to be loaded results from an SQL expression and does not match a column from the table to load. The LOAD statement is not usable as it is specified.

User response: Modify the LOAD statement.

Return Code: 4

**INZU214E LOAD STATEMENT ERROR : FAILED
TO OPEN FILE *syspunch***

Explanation: LSCX502****WARNING****ERRNO = ESYS

Generated in AFOPEN called from line 1881 of @@312162(IRLOAD), offset 000ABC

Extended name: _inzrloadOpenSysin

System macro "OBTAIN" failed with return code 24

Interrupted while: Opening file *syspunch*

This occurs if one of the following data set handled by the utility SYSIN *ddname* used for LOADDDN has a DATACLAS with one of the following attribute:

- DATA SET NAME TYPE
- EXTENDED REQUIRED
- EXTENDED PREFERRED

Local Fix: Consider using another dataclass that does not have any of the above mentioned attributes

**INZU215E LOAD STATEMENT ERROR : INVALID
CONVERSION FOR COLUMN *col_name***

Explanation: This message is issued when LOAD SYSIN GENERATION is requested. An invalid conversion is requested between the data type of the column in the unloaded file and the data type of the column in the table. The LOAD SYSIN cannot be generated.

User response: Modify the output field type of the UNLOAD SYSIN.

Return Code: 8

**INZU216I LOAD STATEMENT CAN NOT BE
GENERATED IN FORMAT CSV**

Explanation: This message is issued when LOAD SYSIN GENERATION is requested. The LOAD SYSIN cannot be generated in format CSV (UNLOAD PLUS syntax).

User response: No action is required.

Return Code: 0

**INZU217W LOAD STATEMENT WARNING :
UNABLE TO COMPUTE SORTKEYS
FOR SELECT AT POS (*line, column*)
SORTKEYS IS SET TO 0**

Explanation: This message is issued when LOAD SYSIN GENERATION is requested. Option SORTKEYS is requested by an unsupported SELECT statement and cannot be computed or EXECUTE NO is specified for this unload. The default SORTKEYS value is 0.

User response: If necessary, manually change the

SORTKEYS value in LOAD SYSIN.

Return Code: 4

INZU218I UNABLE TO GENERATE LOAD STATEMENT FOR DELIMITED FORMAT IN SELECT STARTING AT (line, column)

Explanation: This message is issued when LOAD SYSIN GENERATION is requested. In format COMMA-DELIMITED, syntax FAST UNLOAD, the LOAD SYSIN cannot be generated.

User response: No action is required.

Return Code: 0

INZU219I PTFLEVEL=aaaaaa-nnnnnn

Explanation: This is an informational message that displays the number of the latest APAR (nnnnn) that was applied on the INZUTILB module. The nnnnn value, if any, is intended for internal IBM use only.

User response: No action is required.

Return Code: 0

INZU220E THE SETTINGS FOR DB2 ID IN THE PARMLIB ARE INCOMPLETE OR ARE MISSING

Explanation: DB2 HPU could not find any value of VZD007/DSNEXIT associated with the subsystem or group name DB2 ID that is specified within the PARM of the EXEC card.

User response: Check that the DB2 ID that is specified within the PARM of the EXEC card is a valid DB2 subsystem or group name. Verify that one of the VZD001 parameters was set to DB2 ID and that the VZD007/DSNEXIT associated parameter is present.

Return Code: 8

INZU221I PARALLELISM ACTIVATED FOR PARTITION PROCESSING. MAXIMUM DEGREE OF PARALLELISM IS SET TO n

Explanation: Parallelism of LDS processing was activated by indicating the parallelism degree. This message is issued when unloading a partitioned table space with parallelism or when several SELECT statements that were processed by using SQL are specified in the same UNLOAD block.

User response: No action is required.

Return Code: 0

INZU222I ddname, TOTAL NUMBER OF RECORDS WRITTEN n

Explanation: This message indicates the total number of records that were written into an output file.

User response: No action is required.

Return Code: 0

INZU223I PARALLELISM ACTIVATED FOR PARTITION PROCESSING. MAXIMUM DEGREE OF PARALLELISM IS LIMITED TO n BY PARMLIB PARAMETER VUX005/MAXSORT

Explanation: Parallelism of LDS processing was activated by indicating the parallelism degree. A parallelism parameter was specified in SYSIN but could not be used because SORT had to be activated. The maximum degree of parallelism was then limited by parameter VUX005/MAXSORT.

User response: No action is required.

Return Code: 0

INZU224I IBM DB2 HIGH PERFORMANCE UNLOAD Vn.n

Explanation: This informational message contains the DB2 HPU version and release number.

User response: No action is required.

Return Code: 0

INZU225I PARALLELISM ACTIVATED FOR TABLESPACE PROCESSING. MAXIMUM DEGREE OF PARALLELISM IS SET TO n

Explanation: Parallel processing of the table space was activated because a parallelism degree was specified in the PARALLELISM option or in the PARMLIB parameter VUX030/UTLPARAL. This message is issued when multiple table spaces with parallelism are unloaded.

User response: No action is required.

Return Code: 0

INZU226I PARALLELISM ACTIVATED FOR SELECT PROCESSED BY DB2. MAXIMUM DEGREE OF PARALLELISM IS SET TO n

Explanation: Parallel processing for DB2-processed SELECT statements was activated because a parallelism degree was specified in the PARALLELISM option or in the PARMLIB parameter VUX031/DB2PARAL. This

message is issued when multiple SELECT statements are processed by DB2.

User response: No action is required.

Return Code: 0

INZU227I OPTION *option value* FOR FORMAT *format* WILL BE IGNORED BY UTILITY

Explanation: The option for the SELECT statement is ignored for the specified format.

User response: No action is required.

Return Code: 0

INZU228E *text* EXPRESSION IS NOT SUPPORTED BY UTILITY

Explanation: The SELECT that is coded in SYSIN contains an SQL expression (*text*) which is not natively supported by DB2 HPU, and the SYSIN specified DB2 NO. The SELECT cannot be processed natively.

User response: Specify DB2 YES in SYSIN and rerun the job.

Return Code: 8

INZU230E LOAD STATEMENT ERROR OPTIONS *list of options* ARE NOT COMPATIBLE

Explanation: Incompatible load options are specified for LOAD SYSIN GENERATION.

User response: Modify the LOAD options.

Return Code: 8

INZU231E COPYDDN CHECK ERROR: GENERIC DDN CHECK IS NOT SUPPORTED

Explanation: The COPYDDN option CHECK is used with a generic ddname (for example, DDNIN*).

User response: Remove the CHECK option and specify that the image copy is INLINE by using the INLINE keyword.

Return Code: 8

INZU232E PARAMETER IS INCONSISTENT WITH ITEM AT (*line, column*) ENDING AT (*line, column*)

Explanation: The parameter that was specified in the previous message has a type that is inconsistent with the parameter that is specified in this message.

User response: Change the syntax to eliminate the conflict. Be as specific as possible.

Return Code: 8

INZU233W LOAD STATEMENT WARNING: PARMLIB PARAMETER VUU039/UNLLDER IS SET TO IGNORE. LOAD STATEMENT ERRORS FOR SELECT STARTING AT (*line, column*) ARE IGNORED. LOAD STATEMENT IS BYPASSED

Explanation: A LOADDDN keyword requested a LOAD SYSIN generation but an error prevents it from generating a correct LOAD statement for one specific SELECT because PARMLIB parameter VUU039/UNLLDER is set to IGNORE. Processing continues.

User response: No action is required.

Return Code: 4

INZU234E COPYDDN NOT FOUND FOR PARTITION *i*

Explanation: When unloading a partitioned image copy with parallelism, this message indicates that the selected partition has no COPYDDN in JCL.

User response: Allocate the missing COPYDDN for the indicated partition in the JCL and rerun the job.

Return Code: 8

INZU237I IN *unload / select n* PARTITION *p* CANNOT BE UNLOADED IN DDN *ddnxxp* DDN *ddnxxp* IS ALSO USED AS A BASE DDN IN *select/unload n* PARTITION WILL BE UNLOADED IN BASE DDN *ddn*

Explanation: A suffixed DDN that is used to unload a partition with parallelism is also being used as a base DDN by either a SELECT or an UNLOAD statement. To avoid mixing of data, the partition is unloaded in the base DDN instead of the suffixed DDN.

User response: Check the output data distribution to verify that it is correct. If necessary, change the base ddnames to avoid mixing the ddnames.

Return Code: 0

INZU238E UNEXPECTED UNSUPPORTED SELECT UTILITY CANNOT SWITCH TO SQL PROCESSING REASON *return_code / reason_code*

Explanation: A supported SELECT statement cannot be processed natively by DB2 HPU. This is an unexpected internal error.

User response: Contact IBM Software Support, and provide the content of the PARMLIB, including all members or at least the INZUTIL member and the INZDSSID member, where SSID is the name of the DB2 subsystem or group attach name on which the

INZU239I • INZU256I

issue occurs, the complete job log, and the DDL of the unloaded object.

Return Code: 8

INZU239I INVALID CONVERSION SPECIFIED FOR COLUMN NO *no_column_in_select*, CONVERSION IS IGNORED

Explanation: This message indicates that an invalid conversion is specified on a column containing binary data. For example, internal numeric, internal date-time, binary, or blob. The conversion is ignored.

User response: No action is required.

Return Code: None.

INZU240I PARMLIB VARIABLE VUU058/ULIDENT SET TO 'IGNORE' DUE TO DB2 VERSION PRIOR TO V9

Explanation: The PARMLIB parameter VUU058/ULIDENT contains OVERRIDE, which can only be accepted in DB2 Version 9 or later. This parameter is changed to IGNORE for the current execution.

User response: No action is required.

Return Code: None.

INZU242I UNLOAD OF TABLESPACE *dbname.tsname* IS SPLIT DUE TO THE MAXPART *n* OPTION

Explanation: The MAXPART parameter was used to unload the indicated table space.

User response: No action is required.

Return Code: None.

INZU243E IN SELECT STARTING AT (*line, column*) TABLESPACE *ts_name.db_name* MAXPART *n* AND ORDER BY REQUIRE AN OUTPUT FILE PER PARTITION

Explanation: DB2 HPU cannot perform an ORDER BY or ORDER CLUSTER on a partitioned table because of the MAXPART parameter. The MAXPART parameter is allowed only with an ORDER BY statement when each partition is unloaded in a separate output file.

User response: Modify the SYSIN by either removing the ORDER BY or MAXPART specification, or specify an output file per partition, and rerun the job.

Return Code: 8

INZU250E INVALID DATETIME CONSTANT *constant*

Explanation: An invalid datetime constant was specified in the SELECT statement.

User response: Correct the constant in the SYSIN, and resubmit the job.

Return Code: 8

INZU251E INVALID COMMA BETWEEN SELECT STATEMENT

Explanation: An unauthorized comma was found between SELECT statements.

User response: Correct the SYSIN, and resubmit the job.

Return Code: 8

INZU252W LOAD STATEMENT WARNING: INDDN CANNOT BE DETERMINED FOR SELECT STARTING AT (*line, column*) DEFAULT SYSREC IS USED

Explanation: The output data ddname cannot be determined. This message occurs when LOAD SYSIN GENERATION is requested. This situation can happen when using a TEMPLATE for the output data set.

User response: No action is required.

Return Code: 4

INZU255E ESTIMATION OF SPACE ALLOCATION FOR TEMPLATE *template-name* FAILED. UTILITY FAILED TO ESTIMATE SIZE OF TABLESPACE *dbname.tsname*

Explanation: DB2 HPU needs to estimate space allocation for output files that were allocated by using a TEMPLATE statement. This error occurs when the estimates for the space that is required to unload the table fails.

User response: Specify the SPACE parameter in the TEMPLATE definition, and resubmit the job.

Return Code: 12

INZU256I PROCESSING UNLOAD FROM LISTDEF *list-name*

Explanation: This informational message provides the LISTDEF name when an UNLOAD statement is generated from a LISTDEF.

User response: No action is required.

Return Code: 0

INZU257I GENERATING STATEMENTS FROM SELECT STARTING AT POS(*line, column*) USING LISTDEF *list-name*

Explanation: This informational message provides the LISTDEF name and the SELECT position in the SYSIN when a SELECT statement is generated from a LISTDEF.

User response: No action is required.

Return Code: 0

INZU259E LOAD STATEMENT ERROR: GENERATION FAILED FOR SELECT STARTING AT POS(*line, column*)

Explanation: A LOAD statement generation has failed. Subsequent messages will provide more details about the cause of the failure.

User response: Refer to the subsequent messages for more information about the cause of the error.

Return Code: 8

INZU260I GENERATING LOAD STATEMENT FOR SELECT STARTING AT POS(*line, column*)

Explanation: This informational message is issued during the generation of a load statement that corresponds to a LOADDDN keyword.

User response: No action is required.

Return Code: 0

INZU261I PROCESSING GLOBAL LOAD STATEMENT

Explanation: This informational message is issued when generation of a GLOBAL LOAD statement begins. A GLOBAL LOAD statement occurs when several LOAD statements that correspond to several SELECT statements are written in the same output file. Subsequent messages provide more information about which SELECT statements are being written.

User response: No action is required.

Return Code: 0

INZU262I UNLOAD *nnnnn*: TABLESPACE *dbname.tsname*

Explanation: This is an informational message. A physical table space UNLOAD that uses an UNLDDN keyword was requested.

User response: No action is required.

Return Code: 0

INZU263I SELECT *nnnnn*

Explanation: This informational message indicates the SELECT statement that informational or error messages will be issued for. Subsequent messages provide additional information. The sequential number corresponds to the occurrence of the SELECT statement in an UNLOAD statement.

User response: No action is required.

Return Code: 0

INZU264E UNSUPPORTED SELECT WITH LISTDEF *list-name*

Explanation: A LISTDEF and a SELECT statement are not compatible. Some options are restricted when you use a LISTDEF in a SELECT statement. For example, a WHERE clause is not allowed. Refer to "Fast select and fast listdef select blocks syntax and description" on page 173 for more information. This message indicates the name of the LISTDEF. Message INZU010I precedes this message and contains the position of the SELECT.

User response: Modify the SELECT statement, and resubmit the job.

Return Code: 8

INZU265I TEMPLATE OPTIONS LIMIT IS IGNORED BY UTILITY

Explanation: The LIMIT option (used for TEMPLATE switching in DB2 Version 9.1 for z/OS) for TEMPLATE is ignored by DB2 HPU.

User response: No action is required.

Return Code: 0

INZU266E INVALID ORDER BY IN SELECT FROM A LISTDEF

Explanation: A LISTDEF is used in a SELECT statement that specifies an ORDER BY clause. When using a LISTDEF in a SELECT statement, you can use an ORDER CLUSTER clause, but not an ORDER BY clause.

User response: Modify the SELECT statement, and resubmit the job.

Return Code: 8

INZU267I OPTION PART HAS NO EFFECT ON UNLOAD/SELECT GENERATED FROM LISTDEF

Explanation: The PART keyword of the UNLOAD statement has no effect in SELECT or UNLOAD statements that use a LISTDEF. To control partition unloading, use the PARTLEVEL option of the LISTDEF.

User response: No action is required.

INZU268E • INZU277I

Return Code: 0

INZU268E INVALID USE OF UNLDDN
KEYWORD. TABLESPACE NAME OR
LISTDEF MUST BE SPECIFIED

Explanation: The UNLDDN keyword was used without specifying a table space name or a LISTDEF.

User response: Specify a table space name or a LISTDEF in the UNLOAD statement.

Return Code: 8

INZU269E UNDEFINED VARIABLE *var* IN
TEMPLATE *template-name*

Explanation: An undefined variable is used in a DSN TEMPLATE expression.

User response: Correct the variable name or define it with the TEMPLATESET option.

Return Code: 8

INZU270E PHYSICAL UNLOAD AND LOGICAL
UNLOAD ARE MUTUALLY
EXCLUSIVE WHEN USING LISTDEF

Explanation: DB2 HPU cannot process physical UNLOAD and SELECT statements when they both use a LISTDEF in the same UNLOAD command.

User response: Code separate UNLOAD commands, one for the physical unload and one for the SELECT statements.

Return Code: 8

INZU271E INVALID USE OF DDLDDN :
COPYDDN MUST BE SPECIFIED

Explanation: DDLDDN is allowed only with the COPYDDN option.

User response: Correct the SYSIN, and resubmit the job.

Return Code: 8

INZU272E DDL FOR TABLESPACE *dbname.tsname*
WAS NOT FOUND IN *ddname*

Explanation: A CREATE TABLESPACE statement was not found in the DDLDDN file for the specified table space. When a table space is unloaded from image copy with DDLDDN, the DDL file must contain the CREATE TABLESPACE of the unloaded table space.

User response: Check the CREATE TABLESPACE clause in the DDLDDN file or check the name of the table space in the SYSIN.

Return Code: 8

INZU273E ORIGINOBID IS MANDATORY TO
UNLOAD DROPPED MULTI-TABLES
TABLESPACE FROM FIC

Explanation: The ORIGINOBID keyword must be used in the SELECT statements when the table space is multiple tables and the DDLDDN keyword is specified.

User response: Specify the ORIGINOBID in the SELECT statement, and resubmit the job.

Return Code: 8

INZU274I UNLOAD *nnnnn* IS PROCESSED
THROUGH DB2

Explanation: This is an informational message. A SELECT statement is processed through DB2. The UNLOAD number *nnnnn* corresponds to the occurrence of the UNLOAD statement in the SYSIN.

User response: No action is required.

Return Code: 0

INZU275I SELECT *sssss* IN UNLOAD *uuuuu* IS
PROCESSED THROUGH DB2

Explanation: This is an informational message. A SELECT statement is processed through DB2. The UNLOAD number *uuuuu* corresponds to the occurrence of the UNLOAD statement in the SYSIN, and the SELECT number *sssss* corresponds to the occurrence of the SELECT statement in the UNLOAD.

User response: No action is required.

Return Code: 0

INZU276E SPACE ALLOCATION IS REQUIRED
IN TEMPLATE *template-name*

Explanation: DB2 HPU needed to evaluate the amount of data to be stored in the associated data set in order to set the allocation because the SPACE option was not specified in the template definition. However, DB2 HPU could not determine the expected amount of data. This error usually occurs when DB2 HPU processes a logical unload that includes an unsupported SELECT statement.

User response: Check the template definition, define the space allocation, and resubmit the job.

Return Code: 8

INZU277I PROCESSING UNLOAD *nnnnn* FROM
TABLESPACE *dbname.tsname*

Explanation: This is an informational message. DB2 HPU is processing the *dbname.tsname* table space unload.

User response: No action is required.

Return Code: 0

**INZU278I UNLOAD STATEMENT FROM SYSIN
STARTING AT POS(*line, column*)**

Explanation: This is an informational message. A physical unload (using UNLDDN) was requested from a table space name that is explicitly specified in the SYSIN.

User response: No action is required.

Return Code: 0

**INZU279I SELECT STATEMENTS USING
SINGLE TABLE SPECIFICATION**

Explanation: This is an informational message. DB2 HPU is processing a SELECT statement from a table name that is explicitly specified in the SYSIN. Message INZU280I follows this message.

User response: No action is required.

Return Code: 0

INZU280I SELECT *nnnnn* STARTING AT POS(*line, column*)

Explanation: This is an informational message. DB2 HPU is processing a SELECT statement from a table name that is explicitly specified in the SYSIN.

User response: No action is required.

Return Code: 0

INZU281I UNLOAD STARTING AT POS(*line, column*)

Explanation: This informational message provides the UNLOAD command position in the SYSIN. This message is issued for each UNLOAD command that is coded in the SYSIN.

User response: No action is required.

Return Code: 0

INZU282I OUTPUT DDNAME=*ddname*

Explanation: A JCL allocated ddname is used to process a physical or logical unload. This message is issued after message INZU280I or INZU278I.

User response: No action is required.

Return Code: 0

**INZU283E ERROR IN TEMPLATE LIBRARY
*ddname***

Explanation: An error occurred when parsing a TEMPLATE library. Subsequent messages provide the reason of failure.

User response: Refer to the subsequent messages to determine the reason of the failure.

Return Code: 8

INZU284E ERROR IN LISTDEF LIBRARY *ddname*

Explanation: An error occurred when parsing a LISTDEF library. Subsequent messages provide the reason of failure.

User response: Refer to the subsequent messages to determine the reason of the failure.

Return Code: 8

**INZU287I NO TABLE FOUND IN LISTDEF
list-name FOR SELECT STARTING AT
POS(*line, column*)**

Explanation: A SELECT statement from a LISTDEF is coded in the SYSIN, but the generated list does not contain a table. The SELECT statement that uses this LISTDEF cannot generate any SELECT statements.

User response: No action is required.

Return Code: The return code is equal to the PARMLIB parameter VUU024/UNLZLRC if no other SELECT statement or physical unload is defined in the SYSIN.

**INZU288I NO TABLESPACE FOUND IN LISTDEF
list-name FOR UNLOAD STARTING AT
POS(*line, column*)**

Explanation: A physical UNLOAD statement that uses a LISTDEF is coded in the SYSIN, but the generated list does not contain a table space. The UNLOAD statement that uses this LISTDEF does not generate a physical UNLOAD.

User response: No action is required.

Return Code: The return code is equal to the PARMLIB parameter VUU024/UNLZLRC if no other SELECT statement or physical unload is defined in the SYSIN.

**INZU289I PROCESSING UNLOAD *nnnnn* FROM
UNDETERMINED TABLESPACE**

Explanation: An unsupported SELECT statement is processed through DB2. This message is followed by one INZU280I message for each unsupported SELECT statement. All unsupported SELECT statements are processed with the same unload number.

User response: No action is required.

Return Code: 0

**INZU290I TABLESPACE UNLOADED FROM
IMAGE COPY**

Explanation: An UNLOAD is being performed from an image copy. A subsequent message provides the file name of the image copy that was used.

User response: No action is required.

Return Code: 0

**INZU291I TABLESPACE UNLOADED FROM
LAST IMAGE COPY**

Explanation: An UNLOAD is performed from the most recent full image copy of the table space that was found in the SYSIBM.SYSCOPY when COPYDDN LAST_IC was used in the SYSIN.

User response: No action is required.

Return Code: 0

**INZU292I TABLESPACE UNLOADED FROM
IMAGE COPY *n***

Explanation: An UNLOAD is performed from the *n*th most recent full image copy of the table space that was found in the SYSIBM.SYSCOPY, when COPYDDN *n* is used in the SYSIN.

User response: No action is required.

Return Code: 0

**INZU293E UNLOAD FROM IMAGE COPY
STARTING AT POS(*line, column*). NO
OUTPUT DDNAME FOUND FOR
PARTITION *n***

Explanation: An output file is missing that is needed to unload a partitioned image copy with parallelism. The partition for which the ddname is identified. This type of unload requires one distinct file for each unloaded partition.

User response: Add the missing DDNAME.

Return Code: 8

**INZU297I LOGICAL RECORD LENGTH OF THE
OUTPUT FILE *name* IS SET TO 32756.
RECORDS MIGHT BE TRUNCATED**

Explanation: The maximum length of the output record exceeds the physical limit. PARMLIB parameter VUU020/ULTR32K is set to YES and the output file record format is variable.

User response: No action is required.

Return Code: 0

**INZU298E LOGICAL RECORD LENGTH OF THE
OUTPUT FILE *ddname* IS *lrecl* AND
EXCEEDS THE PHYSICAL LIMIT**

Explanation: The maximum length of the output record exceeds the physical limit. This problem usually occurs when a fixed output format is used and when unloading tables in 32KB-page table space.

User response: Use one of the following methods to reduce the length of the output record:

- Use a variable format (VARIABLE ALL, VARIABLE END, DELIMITED).
- Reduce the length of larger columns by forcing the length in USER format or by using the SUBSTR() function.
- Unload only the necessary columns.

Return Code: 8

**INZU299E DUPLICATE COLUMN DEFINITIONS.
USE EITHER INTO-CLAUSE, FORMAT
USER OR LIKE CLAUSE TO SPECIFY
COLUMN OUTPUT FORMAT.**

Explanation: Output column definitions cannot be specified in both the INTO clause and FORMAT USER clause or in the INTO clause with the LIKE clause.

User response: Suppress one of the format definitions, and resubmit the job.

Return Code: 8

**INZU304E UNABLE TO RETRIEVE TABLESPACE
OR DATABASE CCSID FROM THE
DB2 CATALOG**

Explanation: The table space or database CCSID retrieval failed.

User response: Check the SQLCODE and SQL messages in the job.

Return Code: 8

**INZU305E UNSUPPORTED
ROW_VALUE_EXPRESSION**

Explanation: A row_value_expression occurs in a SELECT statement, and DB2 NO is used in the UNLOAD command. DB2 HPU does not natively support the row_value_expression.

User response: Specify DB2 YES or DB2 FORCE, or modify the expression and rerun the job.

Return Code: 8

INZU306E INPUT IMAGE COPY IS NOT
SUPPORTED WHEN UNLOADING
COLUMN TYPE LOB OR XML

Explanation: An unload from COPY with an input DDNAME specified could not be performed because an XML or LOB column is involved. In such a case, only the LAST_IC keyword is supported by DB2 HPU. The COPYDDN ddname option was specified. In such a case and after maintenance for MR0411126733, only the LAST_IC keyword is supported by DB2 HPU.

User response: Specify COPYDDN LAST_IC or COPYDDN -1.

Return Code: 8

INZU307E COPYDDN -n IS NOT SUPPORTED
WHEN UNLOADING COLUMN TYPE
LOB OR XML

Explanation: The COPYDDN -n option was specified for a logical unload requesting a LOB or XML column be unloaded. For the later situation, only the LAST_IC option is supported by DB2 HPU.

User response: Specify COPYDDN LAST_IC or COPYDDN -1.

Return Code: 8

INZU308E NO ELIGIBLE FLASHCOPY WAS
FOUND FOR COLUMN *column_name*
FROM TABLE *qualifier.table_name*

Explanation: Unloading of an XML or a LOB column from FlashCopy image copies was requested but the mandatory set of copies for the indicated column (see the requirements for using COPYDDN LAST_IC or COPYDDN -1 when a LOB or XML column is involved) was not found in the DB2 catalog.

User response: Create a set of FlashCopy image copies that fulfills the requirements for the COPYDDN LAST_IC or COPYDDN -1 feature when a LOB or XML column is involved.

Return Code: 8

INZU309E COPYDDN OPTION ANYTYPE IS NOT
SUPPORTED WHEN UNLOADING
COLUMN TYPE LOB OR XML

Explanation: The requested ANYTYPE option of the COPYDDN feature is invalid since an XML or LOB column must be unloaded. In such a case, only GLOBAL or PARTITIONED keywords are supported by DB2 HPU.

User response: Replace the ANYTYPE option by either GLOBAL or PARTITIONED according to the available set of images copies.

Return Code: 8

INZU310E CCSID *ccsid* IS NOT *ccsid_class* WHERE
ccsid_class IS EITHER SBCS OR MIXED
OR DBCS.

Explanation: A target CCSID with an irrelevant class was specified.

User response: Specify a CCSID from the indicated class.

Return Code: 8

INZU311E CONVERSION INTO DBCS *ccsid* IS
NOT SUPPORTED FOR
COLUMN *column_name*

Explanation: This message indicates that an invalid conversion into a DBCS CCSID was requested.

User response: Specify a SBCS or VBCS CCSID at the target CCSID/

Return Code: 8

INZU351I ***** EXECUTE MODE SET TO NO

Explanation: This message indicates that the unload was performed in EXECUTE NO mode.

User response: No action is required.

Return Code: 0

INZU352I DATA SET GENERATED.
TEMPLATE=*template_name*
DDNAME=*ddname* DSN=*dsname*

Explanation: This message provides information about a data set that was generated from a TEMPLATE in EXECUTE NO mode.

User response: No action is required.

Return Code: 0

INZU353I DATA SET GENERATED FOR
PARTITION *n*
TEMPLATE=*template_name*
DDNAME=*ddname* DSN=*dsname*

Explanation: This message provides information about a data set that was generated to unload data from the partition of a table space. The data set was generated from a TEMPLATE in EXECUTE NO mode.

User response: No action is required.

Return Code: 0

INZU354E AUTOTAG OVERFLOW. IN SELECT STARTING AT POS(*line,col*) MAXIMUM AUTOTAG VALUE *integer* REACHED

Explanation: The generated value for the autotag has reached the maximum limit.

User response: Increase the size of the autotag.

Return Code: 8

INZU355E AUTOTAG ERROR FOR SELECT *num_select* STARTING AT POS(*line,col*). NO AUTOTAG/AUTOTAG CONFLICT. MERGING RECORDS WITH INCONSISTENT AUTOTAG SPECIFICATIONS INTO OUTPUT DDN=*ddd* IS NOT ALLOWED

Explanation: The AUTOTAG option in one SELECT statement is set to a value other than zero, and the AUTOTAG option in a different SELECT statement is set to zero or is not specified. These inconsistent AUTOTAG values are shared by the same output file. This output file is allocated under *ddname ddd*. The location of one of the conflicting SELECT statements is provided in the message.

User response: Ensure that an output data set is the target of SELECT statements with compatible AUTOTAG specifications by either separating the output data sets of the conflicting statements or by making their AUTOTAG specifications consistent. To make the AUTOTAG specifications consistent, either add or remove the AUTOTAG specifications in the UNLOAD statement.

Return Code: 8

INZU356E AUTOTAG ERROR FOR SELECT *num_select* STARTING AT POS(*line,col*). DIFFERENT SIZES MERGING RECORDS WITH INCONSISTENT AUTOTAG SPECIFICATIONS INTO OUTPUT DDN=*ddd* IS NOT ALLOWED

Explanation: Inconsistent sizes of AUTOTAG fields are requested for SELECT statements that have the same output file. The output file is allocated under *ddname ddd*. The location of one of the conflicting SELECT statements is provided in the message.

User response: Ensure that an output data set is the target of SELECT statements with compatible AUTOTAG specifications by either separating the output data sets of the conflicting statements or by making their AUTOTAG specifications consistent. To make the AUTOTAG specifications consistent, specify the same AUTOTAG specifications in the UNLOAD statement.

Return Code: 8

INZU357E IN SELECT POS(*line, col*), SQLPART OPTION IS INVALID FOR TABLE SPACE *dbname.tsname*

Explanation: The SQLPART option is available only on an index-partitioned table space or a table-partitioned table space.

User response: Remove the SQLPART option and run the job again.

Return Code: 8

INZU358E IN SELECT POS(*line, col*), INVALID SQLPART OPTION TABLE SPACE CANNOT BE DETERMINED

Explanation: The SQLPART option is available only for SELECT statements for which tables and table spaces can be determined.

User response: Remove the SQLPART option, or modify the SELECT statement and run the job again.

Return Code: 8

INZU359E IN SELECT POS(*line, col*), INVALID OUTDDN SQLPART OPTION REQUIRED SAME PROPERTIES TO BOTH OUTPUT DDN

Explanation: DDN1 and DDN2 of OUTDDN(DDN1, DDN2) must be built identically. If OUTDDN uses templates, DDN1 and DDN2 must both be templates with the &PART variable to unload partitions in distinct files, or DDN1 and DDN2 must be templates without the &PART variable to unload partitions in one file. If OUTDDN uses files that are allocated in JCL, each partition or range of partitions must be unloaded in the same build as *ddname* (for example, DDN100i, *ddn DDN200i*).

User response: Modify the definition of the output file to ensure that DDN1 and DDN2 are built the same way.

Return Code: 8

INZU360I SELECT STATEMENT PROCESSING THROUGH DB2

Explanation: This message indicates that the SELECT statement is processed through DB2.

User response: No action is required.

Return Code: 0

INZU361I *** ERRORS SUMMARY *******

Explanation: This message begins an error report that is generated when ONDEMAND_RESOURCE_ALLOCATION is set to YES in PARMLIB parameter VUU030/ULOPTNS. The

error report lists all of the tasks for which the unload failed.

User response: No action is required.

Return Code: None.

INZU362E UNLOAD *nnnn* FROM TABLE SPACE *dbname.tsname* FAILED POS(*line* , *column*)" RC = 0*xnnnnnn*

Explanation: This message indicates that an unload that was natively processed by DB2 HPU failed. This unload was skipped to allow the next unload to process.

User response: Check for other DB2 HPU messages or system messages that might explain why the unload failed. Fix the problem, and run the job again.

Return Code: 8

INZU363E UNLOAD *nnnn* FROM DB2 FAILED POS(*line* , *column*)" RC = 0*xnnnnnn*

Explanation: An unload that was processed through DB2 failed. This unload was skipped to allow the next unload to process.

User response: Check for other DB2 HPU messages or system messages that might explain why the unload failed. Fix the problem, and run the job again.

Return Code: 8

INZU364E UNSUPPORTED DEFAULT VALUE FOR COLUMN *colname* IN LIMIT KEY OF PARTITION *part number* OF TABLE SPACE *dbname.tsname*

Explanation: DB2 HPU cannot build the SQL filter because the default value for the LIMIT KEY of a partition is unsupported.

User response: Specify an explicit partition limit for the indicated columns, or remove the SQLPART option.

Return Code: 8

INZU365E SELECT STARTING AT POS(*line*, *column*) CANNOT BE PROCESSED

Explanation: A SELECT statement cannot be processed.

User response: See other DB2 HPU messages that were issued before INZU365E to determine why the SELECT statement cannot be processed. Fix the problem, and run the job again.

Return Code: 8

INZU366I UTILITY RETURN CODE *rc* (REASON CODE *rs*)

Explanation: This informational message contains the utility return and reason code when an error occurs. The reason code is intended to be used only by support personnel for diagnosis if the cause of the error cannot be determined.

User response: Refer to the SYSPRINT for information about the return code. If you cannot determine the cause of the error, contact IBM Software Support and supply the return and reason codes.

Return Code: 8, 12, or 16

INZU368I PHYSICAL AND LOGICAL UNLOAD FROM GLOBAL FIC WITH PARTITION FILTERING WILL BE PROCESSED IN TWO SEPARATE TASKS

Explanation: A physical unload and a logical unload were requested while unloading from a global image copy with partition filtering or using one output file per partition. The physical unload and the logical unload are run separately. The input image copy is read twice to process the UNLOAD command.

User response: No action is required.

Return Code: None.

INZU370I LDS ALLOCATION FAILED FOR [TABLE SPACE|INDEX SPACE] *dbname.spacename* [PART *nn*], DB2 CATALOG IS ACCESSED TO REFRESH LDS NAME

Explanation: Dynamic allocation of the LDS that corresponds to the table space or index space, and partition has failed. DB2 HPU will refresh the IPREFIX information from the DB2 catalog before dynamic allocation is reissued.

User response: No action is required.

Return Code: None.

INZU371I THE LDS NAME HAS NOT BEEN MODIFIED BY THE SWITCH PHASE OF AN UTILITY

Explanation: Message INZU370I was previously issued, but the value of the IPREFIX that was read from the DB2 catalog is identical to the value that was initially read. Dynamic allocation is not reissued.

User response: No action is required.

Return Code: None.

INZU372I OBID REPORT FOR UNLOAD
unload-number

Explanation: This message is part of the OBID report. It contains the number for the unload that unloaded the image copy.

User response: No action is required.

Return Code: None.

INZU373I IMAGE COPY INFORMATION: SSID
ssid, DBID n X'nnnn' PSID n X'nnnn'

Explanation: This message is part of the OBID report. It contains the SSID and the DBID/PSID that identifies the table space.

User response: No action is required.

Return Code: None.

INZU374I *n* OBID(s) ENCOUNTERED IN THE
IMAGE COPY

Explanation: This message is part of the OBID report. It contains the number of OBIDs that were found in the image copy.

User response: No action is required.

Return Code: None.

INZU375I OBID *n X'nnnn'*

Explanation: This message is part of the OBID report. It contains an OBID that identifies a table that was found in the image copy.

User response: No action is required.

Return Code: None.

INZU376I SELECT/UNLOAD *n* [PARTITION *n*]
NUMBER OF RECORDS WRITTEN *n*

Explanation: This message indicates the total number of records that were written for each SELECT statement and physical unload into the output file that is specified in message INZU222I.

User response: No action is required.

Return Code: None.

INZU377E INVALID PADDING SPECIFICATION.
PADDING_STRICT IS SET IN
PARMLIB PARAMETER
VUU030/ULOPTNS

Explanation: An old style of syntax for PADDING OPTIONS was used, but PARMLIB parameter VUU030/ULOPTNS uses only the new style of syntax.

User response: Change the SYSIN to match the

PADDING (SBCS padding [, DBCS padding [, record padding]]) syntax.

Return Code: 8

INZU378I DEPRECATED PADDING SYNTAX:
USING SYNTAX PADDING(
SBCS_PAD, DBCS_PAD) OR FLAG
PADDING_STRICT IN PARMLIB
PARAMETER VUU030/ULOPTNS IS
RECOMMENDED

Explanation: The PADDING 'c'/x'hh'/x'hhhh' syntax was used, and PADDING_STRICT(NO) was specified in PARMLIB parameter VUU030/ULOPTNS to support the PADDING 'c'/x'hh'/x'hhhh' syntax. However, the PADDING 'c'/x'hh'/x'hhhh' syntax will not be supported in later releases.

User response: Consider using the PADDING (SBCS padding [, DBCS padding [, record padding]]) syntax for PADDING to manage the padding of both CHAR and GRAPHIC data types. Change the SYSIN syntax.

Return Code: 0

INZU379E INVALID PADDING CHARACTER 'c' /
X'hh'

Explanation: An invalid SBCS, DBCS, or record padding character was specified.

User response: Change the padding specification. SBCS and record padding characters must be 1-byte string literals, and the DBCS padding character must be a 2-byte string literal. String literals include 'c', X'hh', G'SI_graphic_char_SO', GX'hhhh', and UX'hhhh'.

Return Code: 8

INZU380I DEFAULT OUTPUT FORMAT IS SET
TO *format* DUE TO PARMLIB
PARAMETER VUU045/ULFORMAT

Explanation: This message specifies the default format that will be used by DB2 HPU when FORMAT has not been specified for a SELECT statement.

User response: No action is required.

Return Code: None.

INZU381I THE COPYDDN KEYWORD IS NOT
VALID. DB2 NO MUST BE SPECIFIED.
THIS SYNTAX IS DEPRECATED AND
WILL NOT BE SUPPORTED IN LATER
RELEASES.

Explanation: Data is unloaded from a table space instead of an image copy when COPYDDN is specified with DB2 YES with an unsupported SELECT statement or DB2 FORCE and when COPYDDN_STRICT(NO) is specified in the VUU030/ULOPTNS PARMLIB parameter. Because specifying DB2 YES or DB2 FORCE

and COPYDDN is ambiguous, this syntax is deprecated and will not be supported in later releases of DB2 HPU .

User response: Update the SYSIN to conform to the new syntax.

Return Code: 0

INZU382E THE COPYDDN OPTION AND THE DB2 YES OR DB2 FORCE OPTIONS ARE INCOMPATIBLE.

Explanation: The COPYDDN keyword cannot be specified with DB2 YES or DB2 FORCE when COPYDDN_STRICT(YES) is specified for the VUU030/ULOPTNS PARMLIB parameter. To unload from an image copy, both DB2 NO and the COPYDDN keyword must be specified.

User response: To unload from an image copy, specify COPYDDN and DB2 NO in the SYSIN.

Return Code: 8

INZU383I THE FOLLOWING KEYWORD IS IGNORED IN DB2 CM8*, CM9*, V10 ENFM AND LATER RELEASES: *keyword_name*.

Explanation: The specified keyword is ignored in DB2 10 for z/OS enabling-new-function mode and later releases. The keyword is ignored for one of the following reasons:

QUIESCECAT

In DB2 10 for z/OS enabling-new-function mode and later releases, DB2 HPU reads the DB2 catalog in SQL mode. In SQL mode, quiescing the catalog does not have an effect. DB2 HPU is used as if QUIESCECAT NO was specified.

SQLACCES

In DB2 10 for z/OS enabling-new-function mode and later releases, DB2 HPU reads the DB2 catalog in SQL mode. This mode is the same as SQLACCES YES. Other values are ignored.

User response: To stop receiving the message, remove the deprecated keyword.

Return Code: None.

INZU384I THE DB2 PARAMETER WAS FORCED TO NO BECAUSE COPYDDN WAS SPECIFIED.

Explanation: The DB2 parameter was forced to NO for one of the following reasons:

- A value for the DB2 parameter was not specified in the SYSIN.

- A request was made for data to be unloaded from an image copy by using the COPYDDN parameter.
- COPYDDN_STRICT(YES) was specified for the VUU030/ULOPTNS PARMLIB variable.

User response: No action is required.

Return Code: 0

INZU385I THE UNLOAD PROCESS IS SERIALIZED BECAUSE OF TAPE MANAGEMENT.

Explanation: An unload process is serialized when a logical unload or a physical unload is processed by using output files that are allocated on tape devices. The tape devices are defined by the TAPEUNIT option.

User response: No action is required.

Return Code: 0

INZU386E THE FOLLOWING UNIT IS DUPLICATED IN THE TAPEUNIT OPTION: *unit_name*.

Explanation: The definition of a tape unit or a storage class is duplicated in the TAPEUNIT option.

User response: Remove one of the duplicate definitions.

Return Code: 8

INZU387E THE ORDER BY CLAUSE IS NOT SUPPORTED WHEN DATA IS UNLOADED IN FORMAT INTERNAL. THE TABLE SPACE IS IN BASIC ROW FORMAT, AND THE *column_name* COLUMN OF THE *table_name* TABLE IS AFTER A VARYING-LENGTH COLUMN.

Explanation: The ORDER BY clause cannot be processed in the INTERNAL format because the records of the table space are in basic row format (BRF), and the specified column that is used in the ORDER BY clause is after a variable length column.

User response: No action is required.

Return Code: 8

INZU388W THE INDEX KEY IS NOT SUPPORTED WHEN DATA IS UNLOADED IN FORMAT INTERNAL. THE TABLE SPACE IS IN BASIC ROW FORMAT AND THE INDEX KEY *column_name* COLUMN OF THE *table_name* TABLE IS AFTER A VARYING-LENGTH COLUMN.

Explanation: The ORDER CLUSTER clause cannot be processed in the INTERNAL format because the records

INZU389E • INZU395E

of the table space are in basic row format (BRF), and the specified column that is used in the key of the clustering index is after a variable length column. The ORDER CLUSTER clause is ignored for the SELECT statement. This message is preceded by message INZU044W.

User response: No action is required.

Return Code: 4

INZU389E THE FOLLOWING CLAUSE IS INCOMPATIBLE WITH FORMAT INTERNAL: *clause_name*.

Explanation: The clause cannot be used with the INTERNAL format.

User response: Remove the clause, or unload the data in another type of format.

Return Code: 8

INZU390E FORMAT INTERNAL AND ANOTHER FORMAT CANNOT BE SPECIFIED WITHIN THE SAME UNLOAD COMMAND.

Explanation: The INTERNAL format cannot be used with other format types in the same UNLOAD command.

User response: Use FORMAT INTERNAL for all logical unloads that are processed natively in the same UNLOAD command, or split the unload by using at least two UNLOAD commands.

Return Code: 8

INZU391E FORMAT INTERNAL IS NOT SUPPORTED FOR SELECT STATEMENTS THAT ARE PROCESSED BY DB2.

Explanation: FORMAT INTERNAL cannot be used for a SELECT statement that is processed by DB2.

User response: Specify a different type of format for SELECT statements that are processed by DB2.

Return Code: 8

INZU392E FORMAT INTERNAL IS NOT SUPPORTED FOR THE FOLLOWING REASON: *reason*. **THE TABLE NAME IS** *table_name*.

Explanation: The INTERNAL format cannot be used because of one of the following reasons:

XML OR LOB COLUMN

The specified table contains a LOB column or an XML column.

FIELDPROC ON A COLUMN

A column in the specified table is defined with a FIELDPROC.

User response: Specify a different type of format.

Return Code: 8

INZU393I THE VALUE FOR THE MAXPART PARAMETER IS FORCED TO 1.

Explanation: The MAXPART parameter was forced to 1 to unload the specified table space.

User response: No action is required.

Return Code: 0

INZU394I DATA FROM THE IMAGE COPY WILL BE UNLOADED WITHOUT PARTITION INFORMATION BECAUSE OF THE FOLLOWING REASON: *reason*.

Explanation: An output file per partition has been requested for a table space that was unloaded from an image copy. The structure of the image copy file or the structure of the unloaded table space in the DB2 catalog or in the DDLDDN file does not allow unloading per partition. DB2 HPU cannot generate an output file per partition. All the unloaded rows will be written in the file that corresponds to the first selected partition. The message text indicates one of the following reasons:

- THE IMAGE COPY CONTAINS DATA FROM A NONPARTITIONED TABLE SPACE. THE SPECIFIED TABLE SPACE IS A PARTITIONED TABLE SPACE.
- THE IMAGE COPY CONTAINS DATA FROM A PARTITIONED TABLE SPACE. THE SPECIFIED TABLESPACE IS A PARTITION-BY-GROWTH TABLE SPACE.

User response: No action is required.

Return Code: None.

INZU395E THE IMAGE COPY CANNOT BE USED FOR THIS TABLE SPACE FOR THE FOLLOWING REASON: *reason*.

Explanation: The input image copy file has a structure that is not compatible with the unloaded table space as described in the DB2 catalog or in the DDLDDN file. DB2 HPU cannot unload from such image copy. The message text indicates one of the following reasons:

**THE IMAGE COPY CONTAINS DATA FROM A PARTITIONED TABLESPACE
THE SPECIFIED TABLESPACE IS A NON-PARTITIONED TABLESPACE**

The image copy contains data coming from a partitioned table space and the table space specified in SYSIN is not partitioned.

THE IMAGE COPY CONTAINS DATA FROM A PARTITIONED-BY-GROWTH TABLESPACE THE SPECIFIED TABLESPACE IS A NON-PARTITIONED-BY-GROWTH TABLESPACE

The image copy contains data from a partitioned-by-growth table space and the table space specified in SYSIN is either not partitioned or partitioned differently.

THE IMAGE COPY CONTAINS DATA FROM A PARTITIONED TABLESPACE OF *number* PARTITIONS THE SPECIFIED TABLESPACE IS A PARTITIONED TABLESPACE AND HAS LESS PARTITIONS :

number The image copy contains data from a partitioned table space that has more partitions than the table space specified in SYSIN.

THE IMAGE COPY CONTAINS DATA FROM A PARTITIONED TABLESPACE OF *number* PARTITIONS THE SPECIFIED TABLESPACE IS A PARTITIONED TABLESPACE AND HAS MORE PARTITIONS : *number* AND UNLOAD BY PARTITION HAS BEEN REQUESTED

The image copy contains data from a partitioned table space that has fewer partitions than the table space that is specified in SYSIN and an output file per partition was requested. In this case, only an unload of a subset of the partitions that are in the image copy or an unload into a single output data set is allowed.

THE IMAGE COPY CONTAINS SEVERAL VERSIONS FOR THE TABLE *table_name* WHILE THE TARGET TABLE HAS NO VERSION

The image copy contains data from a partitioned table space that has versions while the target table has no version.

User response: Correct the structure mismatch either by providing another input image copy or by unload another table space. If the structures are compatible (both are partitioned but the table space specified in SYSIN has a bigger number of partition), consider changing the list of partitions to be unloaded or request a single output data set to be created.

Return Code: 8

INZU398E DB2 YES CANNOT BE SPECIFIED FOR A TABLE WITH CONTROLLED ACCESS ON A ROW OR COLUMN. THE TABLE NAME IS *table_name*, AND THE COLUMN NAME IS *column_name*.

Explanation: Access to the specified table is controlled at the row or column level. For such a table, output depends on whether data is accessed natively or by DB2. DB2 YES is restricted because it does not explicitly set the type of access mode to be used.

User response: Specify DB2 NO or DB2 FORCE. For a table whose access is controlled at the column level, consider removing the protected column from the SELECT statement.

Return Code: 8

INZU399E A TABLE WITH CONTROLLED ACCESS ON A ROW OR COLUMN CANNOT BE UNLOADED IN NATIVE MODE. THE TABLE NAME IS *table_name*, AND THE COLUMN NAME IS *column_name*.

Explanation: Access to the specified table is controlled at the row or column level. According to the specified value of VUU062/ULACCTRL, unloading data in native mode (when DB2 NO is specified) is restricted for this table.

User response: Specify DB2 YES or change the setting of the VUUXXX/ULACCTRL parameter to allow data to be unloaded in native mode. For a table whose access is controlled at the column level, consider removing the protected column from the SELECT statement.

Return Code: 8

INZU400I THE FOLLOWING OUTPUT CCSID FOR AN XML COLUMN MIGHT NOT MATCH ENCODING XML STANDARDS: *ccsid*.

Explanation: The chosen CCSID for an XML column might generate XML with an encoding name that is does not match XML standards.

User response: Check the output CCSID for the XML data.

Return Code: 0

INZU401E UNLOADING A VIEW WITH FORMAT INTERNAL IS NOT SUPPORTED.

Explanation: FORMAT INTERNAL cannot be used to unload a view.

User response: Specify a different type of format.

Return Code: 8

INZU402E SELECT WITH COLUMN LIST IS NOT SUPPORTED WHEN DATA IS UNLOADED IN FORMAT INTERNAL.

Explanation: FORMAT INTERNAL cannot be used with a partial selection of the columns of the table.

User response: Specify a SELECT * statement.

Return Code: 8

INZU403E A FLASHCOPY AND A NON-FLASHCOPY CANNOT BE USED AS INPUT FOR A SINGLE UNLOAD STATEMENT. THE NAME OF THE FLASHCOPY IS *flashcopy_name*, AND THE NAME OF THE NON-FLASHCOPY IS *non-flashcopy_name*.

Explanation: A FlashCopy and a non-FlashCopy have been specified as input for a single UNLOAD statement for a partitioned table space. DB2 HPU cannot process both image copies.

User response: Issue one UNLOAD statement with the FlashCopy as input and a separate UNLOAD statement with the non-FlashCopy as input.

Return Code: 8

INZU404E LOB AND XML FILES CANNOT BE UNLOADED ON TAPE.

Explanation: DB2 HPU cannot unload LOB and XML files on tape.

User response: Specify a DASD output file for LOB and XML files.

Return Code: 8

INZU406E THE REQUESTED QUIESCE OF THE TABLESPACE COULD NOT BE PERFORMED. CHECK DB2 STATUS OF THE TABLESPACE AND RELATED INDEXES.

Explanation: The requested QUIESCE utility could not be performed because of the DB2 status of the table space or its related indexes.

User response: Consider removing the QUIESCE request or perform the relevant DB2 operation to remove the restricted status that is preventing the QUIESCE utility from completing. Also, consider setting the VUU028/ULQSCEBH PARMLIB variable to FORCE so that a STOP/START sequence is attempted when the QUIESCE utility fails because of the DB2 status. Doing so enlarges the number of cases where the unload QUIESCE request can be performed.

Return Code: 8

INZU407W SELECT *n* [PARTITION *n*] MAX_EXPECTED_ROWS (*n*) IS LESS THAN THE UNLOADED ROWS (*n*).

Explanation: The number of effective unloaded rows is higher than the number of expected rows that is specified in the MAX_EXPECTED_ROWS keyword. Further processing might be affected because the sort resources or the size of the output data set might be insufficient.

User response: Modify the value of the MAX_EXPECTED_ROWS keyword so that it reaches at least the number of unloaded rows.

Return Code: 4

INZU408I INDEX SCAN ACCESS HAS BEEN DISABLED DUE TO MAX_EXPECTED_ROWS SPECIFICATION.

Explanation: The index scan access to sort rows by using the cluster index has been disabled because the MAX_EXPECTED_ROWS keyword has been specified in the SELECT statement.

User response: If necessary, remove the MAX_EXPECTED_ROWS keyword to enable the index scan access. The index scan access is more efficient than the sort utility, but it depends on the number of output rows.

Return Code: 0

INZU409E INTERNAL ERROR WHEN PREPARING DB2SORT INVOCATION - REASON 0xnnnn

Explanation: DB2 HPU unexpectedly failed when preparing the DB2 Sort utility invocation.

User response: Contact IBM Software Support.

Return Code: 8

INZU410E INSUFFICIENT RESOURCES FOR DB2SORT INVOCATION. THE NUMBER OF SORTS TO PROCESS IS *n* MAXSORT IS SET TO *p* DB2SORT CAN PROCESS A MAXIMUM OF *q* SORTS IN PARALLEL

Explanation: Not enough resources are available to process the requested sorts with DB2 Sort. The number of sorts to be processed in parallel (i.e. $\min(p, n)$) is greater than the number of sorts that DB2 Sort can handle in parallel (*q*).

User response: Reduce the sort parallelism degree by decreasing the MAXSORT parameter value *p* to a value that is lower than *q*. If a partitioned table space is used, consider reducing the unload tasks parallelism by decreasing the first operand in the PARALLELISM keyword or by using the MAXPART keyword.

Return Code: 8

INZU415I DB2 PARAMETER SET TO FORCE DUE TO SQLID KEYWORD

Explanation: The DB2 parameter was forced to FORCE for the following reason: a request was made to change the CURRENT SQLID.

User response: No action is required

Return Code: 0

INZU416E THE SQLID KEYWORD AND THE DB2 YES OR DB2 NO OPTIONS ARE INCOMPATIBLE

Explanation: The SQLID keyword cannot be specified with DB2YES or DB2 NO. To specify an SQLID, both DB2 FORCE and the SQLID keyword must be specified.

User response: To specify an SQLID, specify SQLID and DB2 FORCE in the SYSIN.

Return Code: 8

INZU417I IMAGE COPIES PRIOR TO BRA rba CANNOT BE USED FOR UNLOADING database.tablespace

Explanation: This message informs that a line describing a DROP COLUMN materialization is found for the tablespace. Image COPY created with a smaller rba value cannot be used for UNLOAD. Database, tablespace and RBA are those that are printed in the DB2 message DSNU556I when trying to do a recover with a too old image copy.

User response: If the message INZU042E is also issued, look at the explanation for this message.

Return Code: None

INZU418E THE POSITION OF A LOB/XML COLUMN IN THE SELECT LIST-OF-COLUMNS/* IS INCOMPATIBLE WITH SPANNED YES OPTION, PROCESS CANNOT CONTINUE BECAUSE ENFORCE_COLUMN_ORDER YES WAS SPECIFIED.

Explanation: The SPANNED YES option was requested. This option requires the LOB or XML data to be placed at the end of the result table (defined by the column list of the SELECT clause and by the optional INTO clause). As some LOB or XML columns positions do not meet this requirement, the unload can only be processed if ENFORCE_COLUMN_ORDER NO which allows DB2 HPU to rearrange the columns of the result table. As ENFORCE_COLUMN_ORDER YES applies to the unload in progress, the processing stops.

If INZU417I message has been previously issued, see the related explanation for more details on the origin of the issue.

User response: Use the value NO of the parameter ENFORCE_COLUMN_ORDER, or change the order of selected columns so that the LOB/XML columns are the last columns of the SELECT statement.

Return Code: 8

INZU419I THE POSITION OF A LOB/XML COLUMN IN THE SELECT LIST-OF-COLUMNS/* IS INCOMPATIBLE WITH SPANNED YES OPTION, COLUMNS WERE REORDERED BECAUSE ENFORCE_COLUMN_ORDER NO WAS SPECIFIED.

Explanation: The SPANNED YES option was requested. This option requires the LOB or XML data to be placed at the end of the result table (defined by the column list of the SELECT clause and by the optional INTO clause). As some LOB or XML columns positions do not meet this requirement and ENFORCE_COLUMNORDER NO applies to the unload in progress, DB2 HPU rearranges the columns of the result table to allow the unload to be processed.

User response: None

Return Code: 0

INZU420I SPANNED YES APPLIES TO SELECT STARTING AT (x,x) AND ENDING AT (x,x). OUTPUT FILE HAS RECFM VBS.

Explanation: As the SPANNED YES option is specified for a logical unload (SELECT clause) which result table has LOB data columns, the mandatory VBS RECFM is needed for the OUTDDN output data set. Further processing will make sure this requirement is fulfilled.

User response: None

Return Code: 0

INZU421W SPANNED YES AT POS (x,x) WILL BE IGNORED BECAUSE THERE IS NO LOB DATA IN SPANNED FORMAT

Explanation: The SPANNED YES option is specified for a logical unload which result table has no LOB data. SPANNED YES does not apply in such a case and is therefore ignored.

User response: Consider removing the SPANNED YES option of you actually do not mean to unload any LOB data.

Return Code: 4

INZU422E THE FORMAT REQUESTED FOR THE LAST SELECTED COLUMN, WHICH IS NOT A LOB OR AN XML, IS NOT COMPATIBLE WITH THE SPANNED YES OPTION.

Explanation: The requested format prevents from

unloading data with the SPANNED YES option (i.e. into a SPANNED file).

User response: Select one of the formats supported when SPANNED YES option is requested or remove the SPANNED YES option if it is not intended.

Return Code: 8

INZU423I SPANNED YES AT POS (x,y) IS INCOMPATIBLE WITH clause.

Explanation: This message informs that the requested clause (ex. FORMAT DELIMITED) is not compatible with the SPANNED YES option. As SPANNED option was specified at the GLOBAL level, it is ignored. The place of the clause indicated by this message is given by an immediately preceding INZU010I message.

User response: None

Return Code: 0

INZU424E SPANNED YES AT POS (x,y) IS INCOMPATIBLE WITH clause

Explanation: The requested clause (ex. FORMAT DELIMITED) is not compatible with the SPANNED YES option. As the SPANNED option was not specified at the GLOBAL level, the ambiguous request cannot be performed. The place of the clause indicated by this message is given by an immediately preceding INZU010I message.

User response: Either remove the SPANNED YES option or do not specify the mentioned clause.

Return Code: 8

INZU425E AT POS (x,x) explanation WHEN SPANNED YES OPTION WAS SPECIFIED

Explanation: A situation prevents a logical unload with the SPANNED YES option to be performed.

The failure cause can be one of the following:

1. HFS FILE NOT ALLOWED: The OUTDDN file allocation (either via JCL DD card of TEMPLATE) HFS file.
2. CCSID CONVERSION NOT ALLOWED FOR LOB/XML COLUMN: the unload command requires a CCSID conversion for a LOB or XML column
3. SPECIFY LOBFILE FOR XML COLUMN
4. UTFREQROWS NOT ALLOWED FOR SELECT PROCESSED BY DB2
5. SORT EXTERNAL NOT ALLOWED
6. USE OF BOTH LENGHT REAL AND LENGTHBYTE NO NOT ALLOWED
7. MULTI-BYTES LENGTH INDICATOR NOT ALLOWED

User response: Consider removing the cause of the failure or the SPANNED YES option.

1. do not allocate the OUTDDN file as an HFS file
2. remove the CCSID conversion request that affects the LOB or XML columns
3. specify a REFORMAT or an INTO clause for XML columns to unload XML data as LOBFILES
4. remove the UTFREQROWS option
5. remove the SORT EXTERNAL option
6. remove either the LENGTH REAL or the LENGTHBYTE NO option
7. specify a one-byte NULL indicator field

Return Code: 8

INZU426E SPANNED DATASET *dsname* CANNOT BE USED TWICE FOR SELECT STARTING AT (x,x) ENDING AT (x,x) AND SELECT STARTING AT (x,x) ENDING AT (x,x)

User response: A dataset used as the output of a logical unload (i.e. a SELECT statement) specifying the SPANNED option is also specified at the output dataset for another logical unload. This is forbidden as a spanned dataset cannot be shared by two unloads. A separate dataset must be specified for each dataset to be written in spanned format.

User response: Specify distincts ouput datasets

Return Code: 8

INZU427I LOAD OPTIONS *options* IS TRANSFORMED INTO A COMMENT BECAUSE ITS DISALLOWED BY DB2 WHEN OPTION SPANNED YES IS USED.

Explanation: A LOADOPT clause was specified for an unload request which output is to be written in SPANNED format. The mentioned option specified by the LOADOPT is not supported by the DB2 LOAD utility when the data to be loaded is a file with RECFM=VBS (spanned file). This incompatible option was transformed into a comment in the generated LOAD command.

User response: None

Return Code: None

INZU428E THE UNLOAD COMMAND, INCLUDING SELECT STARTING *position*, REQUESTS TO UNLOAD THE LOB COLUMN *column name* INTO INCOMPATIBLE OUTPUT FORMATS.

Explanation: At least two logical unloads from a single UNLAD command request the same LOB column be unloaded in spanned instream format on

one hand and in a different format on the other hand (i.e. non-spanned in-stream or LOB file).

User response: Separate the two conflicting logical unloads into two distinct UNLOAD commands.

Return Code: 8

INZU429E XMLSET ERROR: INVALID VARIABLE/STRING - string - IN XMLTAG DEFINITION

Explanation: An invalid variable or string is used to define an XML tag.

| **Operator response:** None

| **System programmer response:** Check XML tag definition.

Return Code: 8

INZU430E XMLSET ERROR: DUPLICATED DEFINITION OF VARIABLE - string -

Explanation: An XMLSET variable is duplicated

| **Operator response:** None

| **System programmer response:** Check XMLSET option.

Return Code: 8

INZU431E XMLSET ERROR: VARIABLE string HAS NO VALUE

Explanation: A XML variable in XMLSET statement cannot be substituted.

| **Operator response:** Check XMLSET statement.

| **System programmer response:** None

Return Code: 8

INZU432E XMLSET ERROR: EMPTY TAG FOR string VARIABLE

Explanation: The resulting XML tag in XMLSET statement is empty.

User response: Correct the XMLSET statement.

Return Code: 8

INZU433E ERROR: CLAUSE INTO IS NOT ALLOWED WITH FORMAT XML

Explanation: Unload plus does not allow INTO clause with FORMAT XML.

User response: Remove the INTO clause or choose another format.

Return Code: 8

INZU434E XMLSET ERROR: INVALID VARIABLE - string -

Explanation: An invalid variable is used in XMLSET statement. Allowed variables are COLUMN, RECORD

User response: Correct the XMLSET options.

Return Code: 8

INZU435E NO LOB DATA ARE TOO LONG TO BE UNLOADED IN SPANNED FORMAT

Explanation: The length of non-LOB data exceeds the maximum supported value. This mainly occurs when unloading data from table in 32 kB page table spaces, or when a LOB column is converted into the VARCHAR data type

User response: Consider one or more of the following options to reduce the length of the non-LOB data:

- Reduce the length of larger columns by specifying a smaller length (if the USER format is used or by adding an INTO clause).
- Use the SUBST() SQL scalar function
- unload only the necessary columns

Return Code: 8

INZU439W MULTI ROW FETCH (*rows number*) HAS BEEN FORCED DUE TO QUERY ACCELERATION REQUEST.

User response: The QUERY_ACCELERATION parameter is specified with a value different from NONE and the UNLEOWSET or its default VUU035/UNLROWSET is set to 1. In this case, the multi-row fetch mode is forced with the number of rows (*rows number*) specified in the message.

User response: Consider specifying ULROWSET to the value mentioned in the message if you do not want the warning message and its related return code 4 be issued.

Return Code: 4

INZU440W QUERY ACCELERATION IGNORED IN THE CURRENT DB2 VERSION

Explanation: The QUERY_ACCELERATION parameter is specified but it is ignored because the current DB2 version is prior to V10 NFM.

User response: Consider removing the QUERY_ACCELERATION option if you do not want the warning message and its related return code 4 be issued.

Return Code: 4

INZU441I PART OPTION IS IGNORED AS A DATA-UNLOAD PROCESS USE SQL MODE

Explanation: The PART keyword is specified but as the request is processed by SQL, the keyword is ignored.

User response: Remove the PART keyword.

Return Code: None.

INZU447E AN ERROR OCCURRED DURING UNLOAD PROCESSING - REFER TO PREVIOUS EXPLANATION MESSAGES

Explanation: An error occurred while unloading the data. The error usually affects only some of the records to be processed. More details about the error is given by previous messages.

User response: Check if the situations reported by the detailed error messages can be solved by changing the unload request.

Return Code: 8

INZU448E SECOND OR THIRD ARGUMENT OF SUBSTR OUT OF RANGE

Explanation: An error was encountered during the SELECT/WHERE clause evaluation. The line concerned was discarded.

User response: Check the SELECT/WHERE clause, SUBSTR function.

Return Code: 8

INZU450I THE IMAGE COPY USED TO UNLOAD THE TABLE *table name* CONTAINS DATA FROM SEVERAL TABLES AND ORIGINOBID 0 WAS SPECIFIED. THIS AMBIGUOUS REQUEST CAN LEAD TO UNPREDICTABLE RESULTS. CHECK THE OBID REPORT GENERATED IN SYSPRINT AND SPECIFY THE OBID OF THE TABLE TO BE UNLOADED.

Explanation: A request to unload the data from an image copy has been done. DB2 HPU has detected that the image copy contains the data of several tables. As ORIGINOBID 0 has been specified, DB2 HPU unload the data from the first table encountered while reading the image copy. As this first table might not be the relevant one, this might lead to unpredictable results. Specifying an explicit value for ORIGINOBID is strongly recommended. In this case, DB2 HPU issues an OBID report which list the OBID of the tables contained in the image copy.

User response: Specify the obid of the table to be unloaded.

Return Code: 0

INZU484I THE VSAM DATA SET DOES NOT EXIST FOR TABLE SPACE *dbname.tsname*

Explanation: Table space *dbname.tsname* was created with DEFINE NO, and DB2 has not defined the underlying LDS files. No rows are unloaded for this object.

User response: No action is required.

Return Code: None.

INZU492W WARNING: DATA FROM COLUMN *column* WAS TRUNCATED TO *n* BYTES FROM *m*

Explanation: Data from a column has been truncated. Truncation occurs for instream BLOB, CLOB, and DBLOB data types to limit the record size to the maximum size allowable.

User response: To avoid truncation, use LOB file reference (datatype BLOBF, CLOBF, DBCLOBF) in a REFORMAT option, INTO clause, or USER FORMAT definition.

Return Code: 4

INZU493W WARNING: CLONE KEYWORD IGNORED : *text*

Explanation: The CLONE keyword has been ignored. The CLONE keyword has no effect when the table space has no clone or when no physical unload (UNLDDN) is requested. The CLONE keyword is supported only for physical unloads. The text field might contain one of the following messages:

- THE TABLESPACE DOES NOT CONTAIN ANY CLONE TABLE
- ONLY SUPPORTED FOR PHYSICAL UNLOAD

To unload a CLONE table using a SELECT statement, specify the name of the CLONE table in the SELECT.

User response: No action is required.

Return Code: 4

INZU494E ERROR: LOBFILE TYPE FOR COLUMN *colname* NOT ALLOWED

Explanation: LOB file data types (CLOBF, DBCLOBF, BLOBF) are supported only for CLOB, DBCLOB, and BLOB columns. You cannot convert any other data type to a LOB file, and you cannot mix a LOB type with a non-corresponding LOB file type (for example you cannot convert a BLOB column into a CLOBF data type).

User response: Check the conversion that is specified in the INTO clause, the REFORMAT option of USER format definition. Correct the invalid conversion, and resubmit the job.

Return Code: 8

INZU495E ERROR: INCORRECT LOBFILE TEMPLATE *template name* FOR COLUMN *col*: . VARIABLE &UNIQ OR USER DEFINED VARIABLE RECNUM IS MISSING

Explanation: A TEMPLATE statement is used to unload LOB data in a LOB file reference, but the DSN template does not contain either the &UNIQ or a user variable set to the :RECNUM predefined variable.

User response: Modify the TEMPLATE definition, and resubmit the job.

Return Code: 8

INZU496E ERROR: INCORRECT OUTPUT TYPE FOR COLUMN *colname*. OUTPUT TYPE MUST BE A CHAR OR VARCHAR LONG ENOUGH (*n* CAR.) TO CONTAIN LOB FILE NAME

Explanation: A LOB file field (subtype BLOBF, CLOBF, or DBCLOBF) is either not a CHAR or VARCHAR, or is not long enough to contain the LOB file reference. This message indicates the required length, which corresponds to the maximum length of the TEMPLATE statement that was used in the LOB file specification, is indicated in the message.

User response: No action is required.

Return Code: 8

INZU497E ERROR: INCORRECT LOBFILE TEMPLATE *template name* FOR COLUMN *col*. VARIABLE &UNIQ OR USER DEFINED VARIABLE RECNUM MUST ONLY BE PART OF THE MEMBER NAME

Explanation: The syntax contains an invalid TEMPLATE specification for a LOB file data type. The &UNIQ variable or a user variable that is set to the :RECNUM predefined variable cannot be used in the name of the PDS. They should be used in the member name.

User response: Modify the template definition, and resubmit the job.

Return Code: 8

INZU498E ERROR: INCORRECT LOBFILE TEMPLATE *template name* FOR COLUMN *col*. VARIABLE &PA. &TSNAME. OR &SPACENAME. MUST BE PART OF THE DATASET NAME

Explanation: The template is invalid for a LOBFILE: because a separate output file is used for each partition. The template that is used for LOB file reference should contain either the partition number (variable &PA.) or the space name (&TS. or &SN. variable).

User response: Modify the template definition, and resubmit the job.

Return Code: 8

INZU499E ERROR: INFORMATION WAS NOT FOUND IN THE *table_name* TABLE FOR THE FOLLOWING *column_type* COLUMN: *column_name*.

Explanation: DB2 HPU did not find information in the specified catalog table for a LOB column or an XML column. The table name is SYSIBM.SYSAUXRELS for a LOB column and SYSIBM.SYSXMLRELS for an XML column.

User response: Determine if the definition of the unloaded object is complete. If the object definition is complete, contact IBM Software Support.

Return Code: 8

INZU501E ERROR: PARTITIONS CANNOT BE UNLOADED IN SEPARATE PDS MEMBERS

Explanation: Unloading a partitioned table space in a PDS file with a separate member for each partition is not supported.

User response: Modify the template definition and resubmit the job.

Return Code: 8

INZU502W WARNING: DATA AFTER CONVERSION CAN BE TOO LONG FOR FIELD *column name*, TABLE *table name*

Explanation: The size of the output field might be too short to contain the converted data when that data is converted in UTF8.

User response: Accept the truncation or increase the length of the output field.

Return Code: 4

INZU503E ERROR: 'ORDER BY' ON LOB OR XML COLUMN IS NOT SUPPORTED

Explanation: You cannot sort output data using an LOB or XML column as sort criteria.

User response: Modify the SYSIN and resubmit the job.

Return Code: 8

INZU504I INFORMATION : COLUMN TYPE *type* IS NOT SUPPORTED NATIVELY

Explanation: A SELECT statement included a column type that DB2 HPU cannot natively unload. If DB2 YES is also specified, the SELECT statement is processed using SQL access.

User response: No action is required.

Return Code: 0

INZU505E ERROR: SORT SIZE CANNOT BE ESTIMATED FOR SELECT STARTING AT POS (*n,n*)YOU MUST SPECIFY THE ESTIMATED NUMBER OF ROWS TO BE SORTED INSTEAD OF 'VSAMSIZE' KEYWORD

Explanation: DB2 HPU cannot estimate the number of rows to be sorted because of the characteristics of the SELECT statement. The VSAMSIZE keyword is accepted only with natively supported SELECT statements that are processed in DB2 FORCE.

User response: Specify the SORT size parameter of the SORT SYSIN KEYWORD or modify the SELECT statement.

Return Code: 8

INZU506W WARNING: EXTERNAL SORT CANNOT BE PROCESSED FOR SELECT STARTING AT POS (*n,n*)THE COLUMN (*column name*) OF THE 'ORDER BY' CLAUSE CANNOT BE FOUND IN THE SELECTED COLUMNS

Explanation: External sort is not supported because a column that was specified in the ORDER BY clause is not in the list of selected columns.

User response: The SORT is processed internally by DB2.

Return Code: 4

INZU507I HIDDEN OPTION IGNORED FOR SELECT PROCESSED BY DB2 STARTING AT POS (*n,n*)

Explanation: The hidden columns cannot be unloaded when the SELECT statement is processed by DB2. The HIDDEN option is ignored.

User response: No action is required.

Return Code: 0

INZU508I TABLE *creator name* HAS BEEN LOCKED IN SHARE MODE DURING UNLOAD PROCESS

Explanation: The LOCK YES parameter changed the access to the specified table.

User response: No action is required.

Return Code: 0

INZU509W UNIT NAME SPECIFIED IN TEMPLATE *templatename* IS IGNORED WHEN USING LOB FILE REFERENCE VARIABLES IN SQL

Explanation: The TEMPLATE definition for *templatename* specifies a UNIT name. This template is used to generate a LOB file reference by using a file reference variable that is processed by DB2. In such a case, the UNIT name is not used to create the file that is pointed to by the file reference. To create the file on a specific unit, use DFSMS ACS routines to assign a unit name based on the dsname of the output file.

User response: Modify the TEMPLATE definition and resubmit the job.

Return Code: 4

INZU510W WARNING: EXTERNAL SORT CANNOT BE PROCESSED FOR SELECT STARTING AT POS (*line, col*)SORT KEY EXCEEDS 4088 BYTES LENGTH

Explanation: An ORDER BY clause with a sort key that is longer than 4088 bytes was specified in a SELECT statement. SORT EXTERNAL is ignored and the SORT is processed by DB2.

User response: No action is required.

Return Code: 4

INZU511W WARNING: EXTERNAL SORT CANNOT BE PROCESSED FOR SELECT STARTING AT POS (*line, col*)THE ORDER BY CLAUSE CONTAINS AN EXPRESSION

Explanation: SORT(EXTERNAL) is ignored and the

ORDER BY clause is processed by DB2.

Return Code: 4

INZU512I IN SELECT STARTING AT POS(*line*,
col), SORT(EXTERNAL) OPTION IS
SET BY UTILITY TO PERFORM
ORDER BY

Explanation: The utility set the SORT(EXTERNAL) option to ORDER BY a SELECT statement because an SQL-partitioned UNLOAD with an ORDER BY statement is performed in a single output file. The SORT(EXTERNAL) option allows sorting data after SQL unloads partitions.

User response: No action is required.

Return Code: None.

INZU513E 'IFERROR NULL' CANNOT BE
SPECIFIED FOR THE COLUMN *name*.
THERE IS NO NULL INDICATOR IN
THE OUTPUT FILE FOR THIS
COLUMN

Explanation: A column format option that specifies IFERROR NULL is defined for an output field that does not contain a null-byte indicator.

User response: Either specify NULLBYTE YES to force the null-byte indicator, or specify a different value in the IFERROR option.

Return Code: 8

INZU515E EXPRESSION WITH DECFLOAT DATA
CANNOT BE NATIVELY PROCESSED
WITHOUT DECIMAL-FLOATING-
POINT FACILITY HARDWARE

Explanation: An expression that uses a DECFLOAT operand cannot be processed natively by DB2 HPU because the Decimal-Floating-Point Facility is not available on the current processor.

User response: Modify the expression in the SELECT statement to allow native processing by DB2 HPU, or specify DB2 FORCE to use DB2 processing.

Reason Code: 8

INZU516E NO CONSISTENT (SAME RBA) FULL
IMAGE COPY FOUND FOR EACH
REQUESTED PARTITION OF THE
TABLE SPACE *dbname.tsname*

Explanation: The CONSISTENT keyword was used, but DB2 HPU did not find consistent input full image copies for the unloaded partitions.

User response: If a consistent image copy is not necessary, remove the CONSISTENT keyword. If a consistent input image copy is necessary, contact your

DBA to ensure that such copies are created for the unloaded object. If you suspect a problem in DB2 HPU, contact IBM Software Support.

Return Code: 8

INZU517E NO FULL IMAGE COPY FOUND FOR
EACH REQUESTED PARTITION OF
THE TABLE SPACE *dbname.tsname*

Explanation: The PARTITIONED keyword was used, but DB2 HPU could not find a partitioned full image copy for each unloaded partition.

User response: Ensure that a copy data set is available for each partition to be unloaded, or remove the PARTITIONED keyword from the UNLOAD statement, if unloading from a global image copy is appropriate.

Return Code: 8

INZU518E THE *template_name* XML TEMPLATE
FOR THE *column_name* COLUMN IS
INCORRECT. THE &PA. VARIABLE
MUST BE PART OF THE PDS OR THE
LIBRARY NAME.

Explanation: When XML data is natively unloaded from a partitioned table space in a PDS or PDSE, the template definition must contain the &PA. variable. When XML data is unloaded by using DB2 (DB2 FORCE), &PA. is not required.

User response: Correct the template definition, and rerun the job.

Return Code: 8

INZX005 PARTITION NO. *nnnn* IS BEING
UNLOADED

Explanation: A partition is being unloaded.

User response: No action is required.

Return Code: 0

INZX006 TABLESPACE UNLOAD PHASE
STARTED

Explanation: The TABLESPACE UNLOAD phase started.

System action: Utility processing continues normally.

User response: No action is required.

Return Code: 0

INZX007 TABLESPACE UNLOAD PHASE
ENDED, *nnnn* ROWS PROCESSED

Explanation: This message indicates normal completion of the TABLESPACE UNLOAD phase and shows the number of rows processed.

INZX009 • INZX089

User response: No action is required.

Return Code: 0

INZX009 PARTITION NO. *nnnn* UNLOADED,
nnnn ROWS PROCESSED

Explanation: The specified partition was unloaded, and *nnnn* records were processed.

User response: No action is required.

Return Code: 0

INZX062 SORT ACTIVATED VIA E15

Explanation: The SORT phase was activated by using input exit E15.

User response: No action is required.

Return Code: None.

INZX063 SORT ACTIVATED VIA SORTIN

Explanation: The SORT phase was activated by using a dynamically allocated SORTIN file.

User response: No action is required.

Return Code: None.

INZX064 SORT ACTIVATED VIA CA-SRAM

Explanation: The SORT phase was activated by using the CA-SRAM interface.

User response: No action is required.

Return Code: None.

INZX068 *sort-identifier* SORT FAILED, RC=*return code*, *sort-parameters*, L=*nnn*

Explanation: The sort for *sort-identifier* failed.

sort-identifier

Specifies (in most of the cases) the name of the table space or index space which rows are sorted. In some cases, *sort-identifier* corresponds to an internal identifier.

return code

Specifies the return code that was issued by the SORT utility.

sort-parameters

Specifies the parameters that were used when activating the SORT utility.

nnn

Specifies the length of the records to be sorted. For example, **INZX068 T1SQXXX-04 SORT FAILED, RC=16, FILSZ=E616068, DYNALLOC=(SYSDA,2), L=107.**

User response: Examine the job output, and determine the cause of the error.

Return Code: Abend.

INZX073 ONE OR MORE PAGES IN ERROR
WERE FOUND, RC = *nn*

Explanation: DB2 HPU encountered one or more pages in error, which it could not correct.

User response: Examine the job output, and determine the cause of the error.

Return Code: 8

INZX081 IMAGE COPY IS BEING READ FROM
ddname or *table_space_name* FLASHCOPY
IS BEING READ

Explanation: DB2 HPU is reading an image copy of the table space from a file that is identified either by the *ddname* or from a FlashCopy.

User response: No action is required.

Return Code: None.

INZX089 *n* RECORDS WRITTEN [IN
hh:mm:ss][,INTERVAL *hh:mm:ss*],
UNLOAD CONTINUES.

Explanation: PARMLIB parameter VUU050/ULFRQMSG is set to a value other than zero. The current number of unloaded rows is indicated by *n*. This message is issued each M rows, where M is the value of PARMLIB parameter VUU050/ULFRQMSG. The message can be prefixed by the system time when the message is written in SYSPRINT.

Depending on the content of PARMLIB parameter VUU050/ULFRQMSG, the following optional information can be displayed:

IN *hh:mm:ss*:

Indicates the elapsed time since the first record was unloaded. This information is displayed if TOTAL is specified in PARMLIB parameter VUU050/ULFRQMSG or if neither TOTAL nor DELTA is specified.

INTERVAL *hh:mm:ss*:

Indicates the elapsed time since the previously issued message or since the first record was unloaded. This information is displayed if DELTA is specified in VUU050.

User response: No action is required.

Return Code: None.

INZX090 *n* RECORDS WRITTEN [IN
hh:mm:ss][,INTERVAL *hh:mm:ss*],
UNLOAD DONE.

Explanation: At the end of the unload process, this message displays the total number of rows that were unloaded in each output file. The message can be prefixed by the system time when the message is written in SYSPRINT.

Depending on the content of PARMLIB parameter VUU050/ULFRQMSG, the following information is also displayed:

IN *hh:mm:ss*:

Indicates the elapsed time since the first record was unloaded. This information is displayed if TOTAL is specified in PARMLIB parameter VUU050/ULFRQMSG or if neither TOTAL nor DELTA is specified.

INTERVAL *hh:mm:ss*:

Indicates the elapsed time since the previously issued message or since the first record was unloaded. This information is displayed if DELTA is specified in VUU050.

User response: No action is required.

Return Code: None.

INZX091 OUTPUT ERROR CODE *return-reason*

Explanation: This message is issued for informational purposes only.

System action: Processing continues.

User response: No action is required.

Return Code: None.

INZX102 INDEX SCAN PERFORMED, *nnnn*
RIDS EXTRACTED

Explanation: This message is issued to inform you about the situation described in message INZX091. *nnn* is the number of RIDs sequentially read from the index. Leaf pages of the index that are misplaced or overflowed are not read.

System action: Processing continues.

User response: No action is required.

Return Code: None.

INZX103 INDEX SCAN STARTED FOR
PARTITION *nnn*

Explanation: This message is issued for informational purposes only.

System action: Processing continues.

User response: No action is required.

Return Code: None.

INZX104 INDEX PARTITION *nnn* SCANNED,
ppp RIDS EXTRACTED

Explanation: This message is issued for informational purposes only.

System action: Processing continues.

User response: No action is required.

Return Code: None.

INZX110 ESTIMATED SIZES (ROWS): *nnnn*

Explanation: This message is issued for informational purposes only.

System action: Processing continues.

User response: No action is required.

Return Code: None.

INZX111 *nnnnK / ppppK* REQUESTED ON *uu*
UNIT(S), SIZE=(*aaaa * bbbb*)

Explanation: This message is issued for informational purposes only.

System action: Processing continues.

User response: No action is required.

Return Code: None.

INZX133 NO VALID INPUT *ddname* *ddname*
FOUND, PARTITION *nnn* NOT
UNLOADED

Explanation: During the unload processing of partition *nnn* of a partitioned table space, DB2 HPU was unable to find the data set with the *ddname* that corresponds to the image copy of partition *nnn*. The partition has not been unloaded and no row has been written into the output file.

User response: This situation might be intentional. Otherwise check that the *ddname* corresponding to the image copy of partition *nnn* has been allocated in the JCL. When unloading the partitions of a table space into independent data sets, one image copy data set per partition must be provided as input data sets. (See "DB2 HPU-allocated ddnames" on page 79.)

Return Code: 4

INZX134 TABLESPACE *tsname* HAS ONLY *number*
LDS. ALLOCATION ERROR ON DATA
SET CAN BE IGNORED.

Explanation: This message is issued to inform that the processing of multiple LDS table space has led to an allocation error message against a non-existing LDS that has been issued and that DB2 HPU has checked

that this was a normal situation considering the actual number of LDS the table space is made of.

This situation occurs when the last LDS of the table space has exactly reached the maximum size of the data set defined for this table space. The reported situation has no impact on the correctness of the unload output.

User response: None.

Return Code: None

INZX206 *tsname* DSNUTILB FAILED RETURN CODE = *nn* ***

Explanation: The DB2 DSNUTILB utility cannot continue because it encountered an error.

User response: Examine the DB2 messages in the SYSPRINT file; see the *DB2 Universal Database for z/OS Messages and Codes* manual for a complete description of possible return codes and reason codes.

Return Code: 8

INZX207 *tsname* DSNUTILB ABENDED ***
*reason code = X'hhhhhhhhh'****

Explanation: The DB2 DSNUTILB utility cannot continue because it abended.

User response: Examine the DB2 messages in the SYSPRINT file; see the *DB2 Universal Database for z/OS Messages and Codes* manual for a complete description of the possible return codes and reason codes.

Return Code: DB2 reason code

INZZ001S (*abend_sequence*) ABEND IN
program_name -
CODE=*system_completion_code*
user_completion_code
REASON=*reason_code*

Explanation: One or more abends occurred, where *abend_sequence*

Indicates the sequence in which this abend occurred during the current process. If a dump is taken for this abend, message INZZ100I(*abend_sequence*) SDWA is issued at the beginning of the dump.

program_name

Indicates the name of the program that was running when the abend occurred. If the program name could not be determined, ***** is displayed.

system_completion_code

Indicates the system completion code.

user_completion_code

Indicates the user completion code.

reason_code

Indicates the hexadecimal reason code.

If a dump is not taken for the abend, this message is followed by either INZZ002I, INZZ003I, or INZZ004I messages. If a problem occurs while the dump is being taken, this message is followed by either INZZ011W, INZZ012W, or INZZ019W messages. If the dump was successfully taken, a message is not issued.

User response: Complete the steps for the following system completion codes:

S000 See the explanation for the user completion code that is indicated in the message text in "DB2 HPU user abend codes" on page 329.

S878 Not enough memory was available to complete the process.

- If IEA705I mentions a problem during FREEMAIN, after you ensure that the VZM009 parameter has the correct value, collect a SYSABEND DUMP if the problem recurs, and contact IBM Software Support.
- If IEA705I mentions a problem during GETMAIN, increase the REGION value or reduce the memory that is necessary for your job by using the instructions in "Decreasing the amount of necessary memory" on page 265.

For other system completion codes, see z/OS MVS System Codes.

Return Code: None.

INZZ002I (*abend_sequence*) NO DUMP REQUESTED

Explanation: A dump was not requested for this abend, so a dump cannot be taken. The abend was specified in the preceding message INZZ001S with the same abend sequence. In the message text, *abend_sequence* indicates the sequence in which this abend occurred during the current process.

User response: No action is required.

Return Code: None.

INZZ003I (*abend_sequence*) NO DUMP PRODUCED FOR THIS SYSTEM CODE

Explanation: The settings indicated that a dump was not to be taken. In the message text, *abend_sequence* indicates the sequence in which this abend occurred during the current process.

User response: If a dump is necessary to diagnose the problem, change the settings so that a dump will be taken, and rerun the job with a SYSABEND DD card.

Return Code: None.

INZZ004I (*abend_sequence*) **SYSABEND
UNAVAILABLE, NO DUMP TAKEN**

Explanation: A dump could not be taken because the SYSABEND ddname was not available. A SYSABEND DD card might not be in the JCL. In the message text, *abend_sequence* indicates the sequence in which this abend occurred during the current process.

User response: Check system messages that might give details about the problem with the SYSABEND ddname. Correct the problem, and rerun the job.

Return Code: None.

INZZ005I (*abend_sequence*) **NO DUMP PRODUCED
DUE TO PREVIOUS ERROR**

Explanation: A dump was not taken. In the message text, *abend_sequence* indicates the sequence in which this abend occurred during the current process.

User response: Check previous messages INZZ004I, INZZ01nx, where *n* is a number and *x* is a character, for the reason why a dump was not taken.

Return Code: None.

INZZ009S (????) **ABEND OCCURED - NO DETAIL
AVAILABLE**

Explanation: An abend occurred and was trapped. However, more detailed information is unavailable. In the message text, ???? indicates that the sequence of this abend could not be determined.

User response: Check other system messages for more information about solving the problem.

Return Code: None.

INZZ011W (*abend_sequence*) **NO DUMP TAKEN -
ABEND DURING OPENING
SYSABEND**

Explanation: An error occurred while opening the SYSABEND file. In the message text, *abend_sequence* indicates the sequence in which this abend occurred during the current process.

User response: Check message INZZ013I and other system messages to determine how to solve the problem.

Return Code: None.

INZZ012W (*abend_sequence*) **ERROR DURING
DUMP, SYSABEND IS TRUNCATED**

Explanation: A failure occurred while the dump was being written. The SYSABEND file does not contain the whole DUMP. In the message text, *abend_sequence* indicates the sequence in which this abend occurred during the current process.

User response: Check message INZZ013I and other system messages to determine how to solve the problem.

Return Code: None.

INZZ013I (*abend_sequence*)
CODE=system_completion_code
user_completion_code
REASON=reason_code

Explanation: This message is issued after message INZZ011W or INZZ012W and explains which abend prevented the dump from being taken correctly.

abend_sequence

Indicates the sequence in which this abend occurred during the current process. If a dump is taken for this abend, message INZZ100I(*number_of_abends*) SDWA is issued at the beginning of the dump.

system_completion_code

Indicates the system completion code of the abend that prevented the dump from being taken correctly.

user_completion_code

Indicates the user completion code of the abend that prevented the dump from being taken correctly.

reason_code

Indicates the hexadecimal reason code of the abend that prevented the dump from being taken correctly.

If a dump is not taken for the abend, this message is followed by either INZZ002I, INZZ003I, or INZZ004I messages. If a problem occurs while the dump is being taken, this message is followed by either INZZ011W, INZZ012W, or INZZ019W messages. If the dump was successfully taken, a message is not issued.

User response: Follow the instructions in the related INZZ011W or INZZ012W messages.

Return Code: None.

INZZ019W (????) **NO DUMP TAKEN - ABEND
DURING OPENING SYSABEND - NO
DETAIL AVAILABLE**

Explanation: A dump could not be taken, and the reason could not be determined. In the message text, ???? indicates that the sequence of this abend could not be determined.

User response: Check other system messages to determine why the abend occurred. If message INZZ021I was issued, the job might remain in a wait state. In this case, consider cancelling the job.

Return Code: None.

INZZ021I (*abend_sequence*) **PROCESSING.
ANOTHER PROCESS IS DELAYED**

Explanation: An abend occurred while another one was being processed. Processing of the new one is postponed. In the message text, *abend_sequence* indicates the sequence in which this abend occurred during the current process.

User response: No action is required.

Return Code: None.

INZZ022I (*abend_sequence*) **PROCESSING.
ANOTHER PROCESS IS WAITING**

Explanation: An abend is being processed while at least one other abend is waiting to be processed. In the message text, *abend_sequence* indicates the sequence in which this abend occurred during the current process.

User response: No action is required.

Return Code: None.

INZZ023I (*abend_sequence*) **PROCESS ENDING.
ANOTHER PROCESS IS RESUMED**

Explanation: The processing of an abend finished, and an abend that had been postponed is about to begin processing. In the message text, *abend_sequence* indicates the sequence in which this abend occurred during the current process.

User response: No action is required.

Return Code: None.

INZZ100I (*abend_sequence*) **SDWA :**

Explanation: This message is written at the beginning of the data that is related to the dump that was taken for one of the abends that occurred during the process. In the message text, *abend_sequence* indicates the sequence in which this abend occurred during the current process.

User response: No action is required.

Return Code: None.

INZ0666F **FATAL ERROR : text**

Explanation: An unexpected exception has occurred. A string might be included at the end of the message. These messages are intended for the IBM Software Support, and are not documented.

User response: Contact IBM Software Support.

Return Code: 16

PLI200 ***** SYSIN FILE COULD NOT BE
OPENED**

Explanation: DB2 HPU was unable to open the SYSIN data set.

User response: Check the JCL that was submitted and the characteristics of the SYSIN data set, if allocated in the JCL, or use Tool Customizer to customize all JCL and data sets that are necessary for the utilities.

Return Code: 12

PLI201 ***** COMMAND OUT OF SEQUENCE**

Explanation: DB2 HPU detected that the specified command is out of sequence.

User response: Check the JCL that was submitted and the characteristics of the SYSIN data set, if allocated in the JCL, or use Tool Customizer to customize all JCL and data sets that are necessary for the utilities.

Return Code: 8

PLI202 ***** ERROR READING SYSIN FILE**

Explanation: DB2 HPU encountered an error when reading the SYSIN data set.

User response: Check the JCL that was submitted and the characteristics of the SYSIN data set, if allocated in the JCL, or use Tool Customizer to customize all JCL and data sets that are necessary for the utilities.

Return Code: 12

PLI203 ***** INVALID COMMAND**

Explanation: DB2 HPU encountered an invalid command.

User response: Check the JCL that was submitted and the characteristics of the SYSIN data set, if allocated in the JCL or use Tool Customizer to customize all JCL, and data sets that are necessary for the utilities.

Return Code: None.

PLI210 ***** REFERENCED MODULE DOES
NOT CONTAIN INZPLIB CSECT**

Explanation: DB2 HPU was unable to locate the INZPLIB CSECT within the referenced module.

User response: Check the JCL that was submitted and the characteristics of the SYSIN data set, if allocated in the JCL, or use Tool Customizer to customize all JCL and data sets that are necessary for the utilities.

Return Code: 8

User response: Contact IBM Software Support, and supply the return and reason codes.

Chapter 8. Reference

These reference topics are designed to provide you with quick access to information about DB2 HPU syntax and data.

Customization reference

Refer to information about DB2 HPU parameters, dsnames, and templates during the customization process.

Topics:

- “Discover process parameters”
- “DB2 HPU library dsnames”
- “DB2 HPU DB2 parameters” on page 397
- “DB2 HPU output data parameters” on page 410
- “DB2 HPU DB2 Administration Tool and DB2 Launchpad parameters” on page 438
- “DB2 HPU conversion parameters” on page 439
- “DB2 HPU file management parameters” on page 440
- “DB2 HPU tuning parameters” on page 447
- “DB2 HPU sort parameters” on page 453
- “Customization templates” on page 455

Discover process parameters

The Discover process parameters on the Discover Customized Product Information panel (CCQPDSC) in Tools Customizer are required to run the Discover EXEC.

You can accept the default values for the Discover process parameters, or you can customize them based on your specific requirements. All Discover process parameters are required.

The following list describes the Discover process parameters.

Old file of variables to be retrieved

The name of the data set that contains the values that you want to retrieve.

Old file with customized file dsnames

The name of the data set that contains the customized dsnames.

Verbose mode for procedure

Specify whether to allow trace information to be retrieved. Specify ON or OFF.

DB2 HPU library dsnames

The DB2 HPU library dsnames section on the Product Parameters panel (CCQPPRD) in Tools Customizer contains the dsnames of the DB2 HPU libraries.

The following list describes the dsnames:

Load modules library (SINZLOAD)

This parameter is required. It specifies the dsname of the load modules library (SINZLOAD).

In previous releases of DB2 HPU, the name of this parameter was VIZ003.

APF load modules library (SINZLINK)

This parameter is required. It specifies the dsname of the APF load modules library (SINZLINK).

In previous releases of DB2 HPU, the name of this parameter was VIZ004.

Samples library (SINZSAMP)

The dsname of the samples library (INZSAMP).

PARMLIB library (INZPLIB/INFPLIB)

This parameter is required. It specifies the dsname of the PARMLIB library (INZPLIB). Do not use the SINZSAMP library. This library is handled by DB2 HPU under the INFPLIB ddname.

Requirement: This library must be the same as the one that is specified in the INZTDSN member. If you do not use this library, DB2 HPU cannot read the customization correctly.

In previous releases of DB2 HPU, the name of this parameter was VIZ007.

Product's DBRM library (SINZDBRM)

This parameter is required. It specifies the dsname of the DBRM library (SINZDBRM).

In previous releases of DB2 HPU, the name of this parameter was VIZ012.

REXX EXEC library (SINZCLST)

This parameter is required. It specifies the dsname of the REXX EXEC library (SINZCLST).

ISPF LOAD modules library (SINZLLIB)

This parameter is required to run the DB2 HPU interactive component. It specifies the dsname of the ISPF load module library (SINZLLIB) that contains the load modules.

In previous releases of DB2 HPU, the name of this parameter was VIZ013.

ISPF messages library (SINZMLIB)

This parameter is required to run the DB2 HPU interactive component. It specifies the dsname of the ISPF messages library (SINZMLIB) that contains the messages.

In previous releases of DB2 HPU, the name of this parameter was VIZ015.

ISPF panels library (SINZPLIB)

This parameter is required to run the DB2 HPU interactive component. It specifies the dsname of the ISPF panels library (SINZPLIB) that contains the ISPF panels.

In previous releases of DB2 HPU, the name of this parameter was VIZ016.

ISPF skeletons library (SINZSLIB)

This parameter is required to run the DB2 HPU interactive component. It specifies the dsname of the ISPF skeletons library (SINZSLIB) that contains the skeletons.

In previous releases of DB2 HPU, the name of this parameter was VIZ017.

ISPF tables library (SINZTLIB)

This parameter is required to run the DB2 HPU interactive component. It specifies the dsname of the ISPF tables library (SINZTLIB) that contains the tables.

In previous releases of DB2 HPU, the name of this parameter was VIZ018.

DB2 HPU DB2 parameters

The DB2 parameters section on the Product Parameters panel (CCQPPRD) in Tools Customizer contains the DB2 parameters that are used by DB2 HPU.

You can accept the default values for the DB2 parameters, or you can configure them based on your specific requirements. Some DB2 parameters are mandatory and must be configured.

The following list describes the common DB2 parameters. The parameters are listed in the following format:

description (Vxxxxxx/parameter-name)

Percentage of displaced pages in inline FIC (VUX023/PGDFIN)

This parameter is optional. It specifies the estimated value of the percentage of displaced pages in inline full-image copies (FIC).

Valid values are 0 - 100. The default value is 20.

The corresponding SYSIN keyword is PGDFIN in the Technical Parameters options block, which can be coded in the GLOBAL OPTIONS block.

In previous releases of DB2 HPU, the name of this parameter was VUX023/PGDFIN.

Use real-time statistics tables for size estimation (VUX036/RTSESTIM)

This parameter is optional. Specify YES to use the real-time statistics tables, when they are available, to estimate the number of rows of the table space to be unloaded. Otherwise, specify NO. This estimate is used to allocate the output data set when a TEMPLATE is used, to set the sort program parameters when an ORDER BY clause is specified, or both.

In previous releases of DB2 HPU, the name of this parameter was VUX036/RTSESTIM.

The default value is NO.

Maximum degree of parallelism for SELECT (VUX031/DB2PARAL)

This parameter is optional. It specifies the parallelism degree for an UNLOAD command when several SELECT statements are processed by using DB2 access (using DB2 FORCE or DB2 YES with unsupported SELECT statements). The value that you specify indicates the maximum number of SELECT statements that are processed in parallel. The value that you specify indicates the maximum number of SELECT statements that are processed in parallel.

In previous releases of DB2 HPU, the name of this parameter was VUX031/DB2PARAL.

Valid values are 1 - 65535. The default value is 5.

Application plan for DB2 HPU (VUM011/PLANOBJT)

This parameter is required. It specifies the plan name for the interactive application and the batch jobs.

Valid values are valid DB2 plan names.

In previous releases of DB2 HPU, the name of this parameter was VUM011/PLANOBJT.

| **Package collection for DB2 HPU (VUM030/COLLOBJT)**

| This parameter is required. It specifies the name of the collection that is used
| to bind the DB2 HPU packages.

| In previous releases of DB2 HPU, the name of this parameter was
| VUM030/COLLOBJT.

| **Owner of the plan created for DB2 HPU (VUM012/PLANOWN)**

| This parameter is required. It specifies the name of the owner of the plan to be
| used to bind the DB2 HPU plan.

| Specify one value for each DB2 subsystem that you defined.

| In previous releases of DB2 HPU, the name of this parameter was
| VUM012/PLANOWN.

| **Grant on the plan created for DB2 HPU (VUX011)**

| This parameter is required. It specifies whether to use the GRANT TO PUBLIC
| or the GRANT TO USER statement to grant privileges to the plan that was
| created for DB2 HPU.

| Specify one value for each DB2 subsystem that you defined with variable
| VZD001.

| Valid values are PUBLIC and USER.

| In previous releases of DB2 HPU, the name of this parameter was VUX011.

| **Quiesce of SYSDBASE and DBD01 for the batch utilities (VUM014/QUIESCAT)**

| This parameter is optional. It specifies whether a quiesce point is to be taken
| on the following table spaces before the job is run.

- | • DSNDB01.DBD01
- | • DSNDB06.SYSCOPY
- | • DSNDB06.SYSDBASE
- | • DSNDB06.SYSDBAUT
- | • DSNDB06.SYSGROUP
- | • DSNDB06.SYSOBJ
- | • DSNDB06.SYSSTATS
- | • DSNDB06.SYSUSER
- | • DSNDB06.SYSVIEWS

| Specify one of the following values:

| **YES**

| A quiesce point is taken at run time unless keyword QUIESCECAT NO
| was specified in the SYSIN of DB2 HPU.

| **NO** A quiesce point is not taken at run time unless keyword QUIESCECAT
| YES was specified in the SYSIN of DB2 HPU.

| **OFF**

| A quiesce point is never taken at run time, even if keyword QUIESCECAT
| YES was specified in the SYSIN of DB2 HPU.

| **FORCE**

| A quiesce point is always taken at run time, even if keyword
| QUIESCECAT NO was specified in the SYSIN of DB2 HPU.

| QUIESCAT is forced to NO in DB2 10 for z/OS ENFM and later releases
| because DB2 HPU accesses the catalog in SQL only at that DB2 level.

The default value is YES.

In previous releases of DB2 HPU, the name of this parameter was VUM014/QUIESCAT.

User who quiesces the catalog table spaces (VUM020/QUIESUSR)

This parameter is optional. It specifies the user who will run the QUIESCE utility on the DB2 catalog table spaces. Specify one value for each DB2 subsystem that you defined with variable VZD001.

Specify one of the following values:

CURRENT_USER

The user who submits the job will be used to run the QUIESCE utility on the DB2 catalog table spaces.

USER name

A specific user name is used to the QUIESCE utility on the DB2 catalog table spaces. The name can be 1 -7 characters.

The default value is CURRENT_USER.

In previous releases of DB2 HPU, the name of this parameter was VUM020/QUIESUSR.

User who quiesces the table space to be unloaded (VUM031/QUITUSR)

This parameter is optional. If a quiesce is requested, this parameter specifies the user who will run the QUIESCE utility on the table space to be unloaded. Specify one value for each DB2 subsystem that you defined with variable VZD001.

Specify one of the following values:

CURRENT_USER

The user who submits the job is used to run the QUIESCE utility on the table space to be unloaded.

USER name

A specific user name is used to run the QUIESCE utility on the table space to be unloaded. The name can be 1 - 7 characters.

In previous releases of DB2 HPU, the name of this parameter was VUM031/QUITUSR.

The default value is CURRENT_USER.

SQL access for reading DB2 catalog (VUM027/SQLACCES)

This parameter is optional. It specifies which of the following three methods is used to access the DB2 catalog:

DIRECT

Direct access on all catalog information

SQL

SQL access on all catalog information

MIXED

Direct access for all information except for the IPREFIX of the SYSINDEXPART and SYSTABLEPART tables (the I/J), which might change after an online reorganization

Specify one of the following values:

MINIMAL

The DB2 catalog is accessed by using DIRECT mode (DIRECT access maximum) and is completed by using the SQL mode (SQL access minimum).

AUTO

The DB2 catalog is accessed either in DIRECT mode or in SQL mode.

YES

The DB2 catalog is accessed only in SQL mode.

NO The DB2 catalog is accessed only in DIRECT mode.

When SQL access is used to read the DB2 catalog, the user ID that is specified in PARMLIB variable VUM032/ACTLGUSR must have SELECT authority on the tables of the DB2 catalog that DB2 HPU accesses.

The access method that is used depends on the SQLACCESS setting and whether QUIESCECAT comes from the PARMLIB or from the SYSIN. The following table shows the access method that is used by DB2 HPU based on the SQLACCESS and QUIESCECAT settings.

Table 29. DB2 catalog access dependencies

SQLACCES value	QUIESCECAT value	Access method used
YES	Any	SQL
NO	Any	DIRECT
AUTO	NO	SQL
AUTO	YES	DIRECT
MINIMAL	NO	MIXED
MINIMAL	YES	DIRECT

For a table space that was not altered recently, using MIXED access with a QUIESCECAT setting of NO guarantees a consistent reading of information from the DB2 catalog, even if an online reorganization (other than an ALTER operation) was recently done. Using MINIMAL provides the best compromise between consistency and speed.

SQLACCES is forced to YES in DB2 10 for z/OS ENFM and later releases because DB2 HPU only accesses the catalog in SQL at that DB2 level.

The default value is MINIMAL.

The corresponding SYSIN keyword is SQLACCES in the Technical Parameters options block, which can be coded in the GLOBAL OPTIONS block.

In previous releases of DB2 HPU, the name of this parameter was VUM027/SQLACCES.

User who runs the DISPLAY command (VUM028/DISPLUSR)

This parameter is optional. It specifies the user who will run the DISPLAY command.

CURRENT_USER

The user who submitted the job is used to run the DISPLAY command. CURRENT_USER is the default value.

USER name

A specific user name is used to run the DISPLAY command. The name can be 1 - 7 characters.

Specify one value for each DB2 subsystem that you defined with variable VZD001.

The default value is CURRENT_USER.

In previous releases of DB2 HPU, the name of this parameter was VUM028/DISPLUSR.

Use DB2 to process unsupported SELECT statements (VUU011/ULSEDB2)

This parameter is optional. It specifies whether DB2 will process SELECT statements when the statements are not supported by DB2 HPU. This parameter is used only if the DB2 option is not specified in the SYSIN. When COPYDDN_STRICT(YES) is specified in the VUU030/ULOPTNS parameter, unloading data from the table by using DB2 instead of unloading data from the image copy is not allowed. Therefore, if you are unloading from an image copy and COPYDDN_STRICT(YES) is specified, the value of the VUU001 parameter is forced to NO.

Specify one of the following values:

NO SELECT statements that are not supported by DB2 HPU will not be processed by DB2.

YES

SELECT statements that are not supported by DB2 HPU will be processed by DB2, unless DB2 NO was specified in the SYSIN of DB2 HPU.

The default value is YES.

In previous releases of DB2 HPU, the name of this parameter was VUU011/ULSEDB2.

Lock the tables in the table space (VUU012/ULLOCK)

This parameter is optional. It specifies whether to lock the tables in the table space.

Specify one of the following values:

NO Tables in the table space are not to be locked unless YES was specified in the SYSIN of DB2 HPU.

YES

Tables in the table space are locked unless NO was specified in the SYSIN of DB2 HPU.

Specifying LOCK NO does not preclude DB2 from taking locks if SQL Access is used to access the data.

The default value is NO.

In previous releases of DB2 HPU, the name of this parameter was VUU012/ULLOCK.

Quiesce the table space (VUU013/ULQSCE)

This parameter is optional. It specifies whether to quiesce the table space.

Specify one of the following values:

NO The table space is not quiesced unless YES was specified in the SYSIN of DB2 HPU.

YES

The table space is quiesced unless NO was specified in the SYSIN of DB2 HPU.

The default value is NO.

Attention: DB2 HPU operates on the physical VSAM data set level that is outside of DB2. If you run DB2 HPU on a table in which a row was just inserted, the unloaded data might not contain the row that was inserted. The unloaded data does not show the row because DB2 might not have externalized the data to DASD yet. This situation can happen when you use DB2 HPU without issuing a QUIESCE (or STOP) on the object. Be careful when using QUIESCE NO.

In previous releases of DB2 HPU, the name of this parameter was VUU013/ULQSCE.

Degree of parallel processing (VUU021/ULDEGREE)

This parameter is applicable only when DB2 extracts data. It specifies the number of parallel tasks or I/O operations that DB2 can use to extract data from a partitioned table space.

Specify one of the following values:

1 Parallelism is not used.

ANY

DB2 HPU decides whether parallelism will be used.

CURRENT_DEGREE

DB2 HPU must keep the default value for the current degree that is set in the DB2 customization. If the PARMLIB variable is set to CURRENT_DEGREE, DB2 HPU will not issue any SET CURRENT DEGREE statements before issuing the SELECT statement.

The default value is CURRENT_DEGREE.

The corresponding SYSIN parameter is ULDEGREE in the GLOBAL OPTIONS block.

In previous releases of DB2 HPU, the name of this parameter was VUU021/ULDEGREE.

Default scheme for UNLOAD TABLESPACE (VUU023/UNLSCHEM)

This parameter is optional. It specifies the unload format for the data.

Specify one of the following values:

ASCII

Specifies that the unloaded data must be in ASCII format. DB2 HPU uses the ASCII CCSID of the subsystem, unless you override it by specifying the CCSID option in the SYSIN of DB2 HPU.

ASIS

Specifies that the data is unloaded in its original format. If the specification for the underlying table space cannot be determined (for example, if the data is processed by DB2), the CCSID that is returned by a standard prepare statement in SQLDA is used. You can also override ASIS by specifying the CCSID keyword.

Specifying ASIS does not mean that a conversion is not required. Conversion might still be required when columns that are not CHAR, VARCHAR, GRAPHIC, and VARGRA types are unloaded in an external format and when the schema of the unloaded table space is not system-EBCDIC.

DB2 HPU first converts the value to the external format (in system-EBCDIC), and the result is converted to the table space schema.

Similarly, the padding characters or field separators (FORMAT DELIMITED) are in system-EBCDIC by default. They are also converted to the table space schema if the table space schema is not EBCDIC.

Conversion is also required when the table space CCSID is not the same as the SYSIN CCSID and when the SELECT statement causes C'constants'.

EBCDIC

Indicates that the data is unloaded in EBCDIC format. DB2 HPU uses the EBCDIC CCSID of the subsystem unless you override it by specifying the CCSID keyword.

UNICODE

Indicates that the data is unloaded in UNICODE format. DB2 HPU uses the UNICODE CCSID of the subsystem unless you override it by specifying the CCSID option.

The default value is EBCDIC.

In previous releases of DB2 HPU, the name of this parameter was VUU023/UNLSCHEM.

Quiesce process (VUU028/ULQSCEBH)

This parameter is optional. It specifies whether the process of updating physical objects in linear data sets (LDS) must be forced or only attempted.

Specify one of the following values:

FORCE

The process of updating physical objects in LDSs is forced, which means that the object is quiesced if possible; otherwise, a STOP/START is forced. The default value is FORCE.

TRY

The process of updating physical objects in LDSs is attempted, which means that the object is quiesced if possible; otherwise, the processing terminated in error.

The default value is FORCE.

In previous releases of DB2 HPU, the name of this parameter was VUU028/ULQSCEBH.

DB2 HPU additional features (VUU030/ULOPTNS)

This parameter is optional. Use this parameter to activate functions for DB2 HPU syntax. This parameter can receive a list of optional parameters, separated by commas, that modify DB2 HPU behavior. Use a continuation comma at the end of each line or fill in the line up to the end.

LOADINDDN(YES)

The INDDN *ddname* card is generated into the LOAD command. The variable *ddname* points to the data set that contains the unloaded data.

When a TEMPLATE is used to allocate the output file, DB2 HPU also generates a TEMPLATE in the LOADDDN file to allocate the input file for the LOAD.

LOADINDDN(NO)

The INDDN card is not generated into the LOAD command. The *ddname* points to the data set that contains the unloaded data.

When a TEMPLATE is used to allocate the output file, DB2 HPU does not generate a TEMPLATE in the LOADDDN file.

DSNTIAULSTRICT(YES)

The SELECT statements that are coded with FORMAT DSNTIAUL will function as if they are coded as FORMAT DSNTIAUL STRICT.

DSNTIAULSTRICT(NO)

The SELECT statements that are coded with FORMAT DSNTIAUL are not changed into FORMAT DSNTIAUL STRICT.

Attention: Changing the behavior of the DSNTIAUL format might affect the content of the unloaded data. Use the DSNTIAUL(YES) parameter and DSNTIAUL(NO) parameter with caution. See “DSNTIAUL block syntax and description” on page 193 for a description of the difference between specifying FORMAT DSNTIAUL and specifying FORMAT DSNTIAUL STRICT.

DELIMITEDGRAPHEXT(YES)

In the DELIMITED format, GRAPHIC and VARGRAPHIC columns are unloaded in ASCII or in EBCDIC as GRAPHIC EXTERNAL data, including the SO/SI characters.

DELIMITEDGRAPHEXT(NO)

In the DELIMITED format, GRAPHIC and VARGRAPHIC columns are unloaded in ASCII or in EBCDIC as GRAPHIC data, without the SO/SI characters.

The DELIMITEDGRAPHEXT parameter is used only for DB2 HPU syntax. It has no impact on Fast Unload or Unload Plus syntax and has no impact when GRAPHIC data is unloaded in UNICODE.

ONDEMAND_RESOURCE_ALLOCATION(YES)

When DB2 HPU processes a list of table spaces by using a LISTDEF and TEMPLATE in a single DB2 HPU step, TEMPLATE files and control blocks are allocated when the table space is processed and are freed afterwards. If you use LISTDEF, you can run DB2 HPU jobs with more table spaces that can be processed in a single DB2 HPU invocation. Specifying YES reduces the memory resources that DB2 HPU uses. If any part of the unload process fails, processing continues with other unload tasks, and the process that failed will be displayed in the list of failed processes that is indicated after message INZU361I.

ONDEMAND_RESOURCE_ALLOCATION(NO)

When DB2 HPU processes a list of table spaces by using a LISTDEF and TEMPLATE in a single DB2 HPU step, all resources are allocated at the beginning of the step. If any part of the unload process fails, processing stops, and DB2 HPU terminates immediately after the error.

PADDING_STRICT(YES)

The previous syntax for the PADDING option that allowed you to code a two-byte hexadecimal constant, such as PADDING x'hhh', is not accepted. The default value for the DBCS padding character is always the DBCS space that was converted in the output CCSID.

COPYDDN_STRICT(YES) or COPYDDN_STRICT(NO)

In previous releases, when both COPYDDN and DB2 FORCE were used, COPYDDN was ignored, and the data was unloaded from the table. When COPYDDN and DB2 YES were used, results could change along with DB2 HPU maintenance. When an unsupported SELECT statement became

supported because maintenance was applied, DB2 HPU unloaded the data from the image copy instead of from the table.

In both cases, the data source that was selected by DB2 HPU might not be the expected one. In later releases, COPYDDN and DB2 YES or DB2 FORCE will be made incompatible to avoid this ambiguous behavior.

You can enable the behavior of DB2 HPU V4.1 and later releases by specifying COPYDDN_STRICT(YES). Otherwise, specify COPYDDN_STRICT(NO), which is the default value.

In later releases, COPYDDN_STRICT(YES) or COPYDDN_STRICT(NO) will be ignored, and only the enhanced behavior, corresponding to the behavior when COPYDDN_STRICT(YES) is specified, will be available.

BY_SQL_ONLY(WITH_UR)

SELECT statements with the WITH_UR clause are always processed in SQL mode. SELECT statements without the WITH_UR clause are always processed in native mode.

GBLPARAL_MAXPART_SET_NULL(YES) or GBLPARAL_MAXPART_SET_NULL(NO)

Processing subsets of partitions (when you use MAXPART or ULMAXPAR when the specified value is less than the number of partitions to be unloaded) and unloading without parallelism because GBLPARAL (NO) is specified are incompatible.

When VUU060/ULMAXPAR *n* and VUU036/GBLPARAL NO are specified together and are applicable to the unload in progress, GBLPARAL NO is ignored, and the data is unloaded without parallelism at the partition level unless GBLPARAL_MAXPART_SET_NULL(YES) is specified.

Specify GBLPARAL_MAXPART_SET_NULL(YES) to disable the processing of the subset of partitions and unload the data without partition parallelism instead. If you specify either the PARALLELISM or the MAXPART keyword in the UNLOAD statement, specifying GBLPARAL_MAXPART_SET_NULL(YES) does not affect parallelism.

AUTO_UNCNT_MAX(*unit_count_value*)

Sets the maximum value to be used as the unit count when the unit count is dynamically determined by DB2 HPU. The value is determined by DB2 HPU when the value is not specified for the UNCNT option in the TEMPLATE statement.

Valid values are 0 - 59.

CHECK_CCSID_STRICT(YES|NO)

Specifies with CCSID classes, such as SBCS or MIXED or DBCS, are allowed.

YES Using an irrelevant CCSID class, such as SBCS or MIXED or DBCS, in the CCSID(ccsid_sbc, ccid_mixed, ccid_dbcs) option is not allowed.

NO Using an irrelevant CCSID class, such as SBCS or MIXED or DBCS, in the CCSID(ccsid_sbc, ccid_mixed, ccid_dbcs) option is allowed.

The default value is YES.

The default values are LOADINDDN(NO), DSNTIAULSTRICT(NO), DELIMITEDGRAPHEXT(NO), ONDEMAND_RESOURCE_ALLOCATION(NO), COPYDDN_STRICT(NO), and BY_SQL_ONLY().

| **INSTREAM_XML_AS_CLOB(YES|NO)**

| Specifies with CCSID classes, such as SBCS or MIXED or DBCS, are
| allowed.

| **INSTREAM_XML_AS_CLOB(YES)**

| XML data unloaded in-stream are unloaded as CLOB data.

| **INSTREAM_XML_AS_CLOB(NO)**

| XML data unloaded in-stream are unloaded as XML unless
| FORMAT DSNTIAUL STRICT is requested in which case the XML
| data is unloaded as CLOB.

| The default value is YES.

| **INSTREAM_XML_AS_CLOB(YES|NO)**

| Specifies with CCSID classes, such as SBCS or MIXED or DBCS, are
| allowed.

| **INSTREAM_XML_AS_CLOB(YES)**

| XML data unloaded in-stream are unloaded as CLOB data.

| **INSTREAM_XML_AS_CLOB(NO)**

| XML data unloaded in-stream are unloaded as XML unless
| FORMAT DSNTIAUL STRICT is requested in which case the XML
| data is unloaded as CLOB.

| The default value is YES.

| **TRUE_UNICODE(YES|NO)**

| **TRUE_UNICODE(YES)**

| When conversion into unicode is requested with no target CCSID
| specification, the CSSID UTF8 is used as a target CCSID.

| **TRUE_UNICODE(NO)**

| When conversion into unicode is requested with no target CCSID
| specification, the default SBCS CSSID of the subsystem is used as a
| target CCSID.

| The default value is NO.

| **CHECK_CCSID_STRICT(YES|NO)**

| **CHECK_CCSID_STRICT(YES)**

| The use of irrelevant CCSID class (i.e. SBCS or MIXED or DBCS) in
| the CCSID (ccsid_sbcs, ccid_mixed, ccid_dbcs) option is
| prohibited.

| **CHECK_CCSID STRICT(NO)**

| The use of irrelevant CCSID class (i.e. SBCS or MIXED or DBCS) in
| the CCSID (ccsid_sbcs, ccid_mixed, ccid_dbcs) option is allowed.

| The default behavior of DB2 HPU is CHECK_CCSID_STRICT (YES).

| In previous releases of DB2 HPU, the name of this parameter was
| VUU030/ULOPTNS.

| **U_NULLCHAR**

| UNLOAD PLUS NULLCHAR parameter default. This parameter is used
| for UNLOAD PLUS syntax and allows you to change the null indicator
| character in the output record.

| Allowed values: xx where xx is any valid hexadecimal constant, except
| X'00'.

The default value is 6F.

U_NULLTYPE

UNLOAD PLUS NULLCHAR parameter default. This parameter is used for UNLOAD PLUS syntax and allows you to specify the location and length of the null indicator field in the output record.

Allowed values: T1/L1/T2/L2

The default value is T1.

DB2 SETTINGS_SOURCE (AUTO|DSNHDECP)

AUTO

The way DB2 HPU determines the DB2 settings depends on the DB2 version. Prior to DB2 V9, the DSNHDECP member from the DSNEXIT library is loaded; in DB2 V9 and later versions, the GETVARIABLE SQL function is used

DSNHDECP

the DSNHDECP member from the DSNEXIT library is used regardless of the DB2 release.

The default value is DB2_SETTINGS_SOURCE (AUTO).

TEMPLATE_SSID_SUBSTITUTION (MEMBER_NAME)

The SSID template variable is substituted with the current DB2 member name to which DB2 HPU has connected.

TEMPLATE_SSID_SUBSTITUTION (GROUP_NAME)

The SSID template variable is substituted with the DB2 member name in a non datasharing environment and with the group attachment name for datasharing environment.

Here is an example that shows how to specify a very long value for ULOPTNS

```
TCUSTMZR           Multiple Value Parameter           Row 1 to 4 of 4
To add another value, issue the ADD command, and specify another value
for the parameter in the blank field in the Value column. Press enter to
save your values, or press End do save and exit.
Commands: ADD
Parameter description:
  DB2 HPU additional features (VUU030/UNLOPTNS)
Line commands: D - Delete
Cmd Value:
  (LOADINDDN (YES),DSNTIAULSTRICT(YES),DELIMITEDGRAPHEX
  T(YES),ONDEMANDE_RESOURCE_ALLOCATION(YES),PADDING_STRIC
  T(YES),COPYDDN_STRICT(YES),BY_SQL_ONLY(WITH_UR),GBLPAR
  RAL_MAXPART_SET_NULL(YES) )
-----End of values-----
Command ==>                               Scroll ==>PAGE
```

Check image copy before unloading (VUU033/ULCHKCPY)

This parameter is optional. It specifies whether the dsname that was specified by the COPYDDN parameter in the SYSIN should be checked.

Specify one of the following values:

|
| **YES**

| The dsname that was provided by the COPYDDN parameter in the SYSIN
| is checked. This PARMLIB parameter is considered if CHECK or INLINE is
| not specified in the SYSIN. The return code is 4, with a warning message
| in the SYSOUT if the check against the SYSIBM.SYSCOPY failed, and the
| FIC corresponding to the dsname is considered as a non-inline FIC.

| **NO** The dsname is not checked.

| The default value is NO.

| In previous releases of DB2 HPU, the name of this parameter was
| VUU033/ULCHKCPY.

| **Option to modify the like behavior in SQL statement (VUU034/ULLIKE)**

| This parameter is optional. It specifies how DB2 HPU handles UNICODE
| MIXED strings in a LIKE predicate.

| Specify one of the following values:

| **STRICT**

| Indicates that SQL rules are used to handle UNICODE mixed strings. To
| handle UNICODE mixed strings by using SQL rules, DB2 HPU must
| convert all operands (columns and masks of the LIKE predicate) in
| UNICODE DBCS, which increases CPU consumption due to conversions
| for each unloaded row.

| **FAST**

| Indicates that UNICODE mixed strings are considered as SBCS strings in
| LIKE predicates. The FAST option avoids MIXED to DBCS conversions and
| improves UNLOAD performances.

| The default value is FAST.

| In previous releases of DB2 HPU, the name of this parameter was
| VUU034/ULLIKE.

| **DB2 HPU response when a LOADDN cannot be generated (VUU038/UNLLDER)**

| This parameter is optional. It specifies whether DB2 HPU will stop when it is
| unable to generate a requested LOADDN. This situation can occur when the
| output format that was specified for DATE, TIME, or TIMESTAMP is not
| supported by the DB2 LOAD utility.

| Specify one of the following values:

| **STOP**

| DB2 HPU stops with return code 8.

| **IGNORE**

| DB2 HPU issues a warning message and continues processing.

| The default value is STOP.

| In previous releases of DB2 HPU, the name of this parameter was
| VUU039/UNLLDER.

| **Unload HIDDEN Column by select * (VUU042/ULHIDDEN)**

| This parameter is optional. It specifies whether hidden columns will be
| selected when a SELECT * statement from a table name or from a LISTDEF is
| used. This parameter applies only to SELECT statements that are processed
| natively. Use the HIDDEN parameter of the EXEC card or the HIDDEN option
| of the OPTIONS block to override this parameter.

Specify one of the following values:

NO Specifies that hidden columns will not be unloaded when a SELECT * statement is used.

YES

Specifies that hidden columns will be unloaded when a SELECT * statement is used. For SELECT statements that are processed by DB2, the HIDDEN option is ignored.

The default value is NO.

In previous releases of DB2 HPU, the name of this parameter was VUU042/ULHIDDEN.

Unload controlled access table in native mode (VUU062/ULACCTRL)

This parameter is optional. Use it to unload data from tables whose access is controlled at the row level or the column level. This parameter applies only when the data of the table is accessed in native mode, which means a physical unload or a logical unload with DB2 NO specified and a supported SQL statement. To unload a row-controlled table, specify DB2 FORCE or specify DB2 NO, and set the VUU062 parameter to YES.

YES Tables that are controlled at the row or column level can be unloaded in native mode.

NO Tables that are controlled at the row or column level cannot be unloaded.

The default value is NO.

In previous releases of DB2 HPU, the name of this parameter was VUU062/ULACCTRL.

Technical parameter to unload XML columns (VUU063/ULOPTLX)

This parameter is optional. Use this parameter to tune the low-level technical algorithm that DB2 HPU uses to unload XML data. If the default optimized values do not apply to your specific case, you can use the VUU063 parameter to enhance performance.

Important: Do not specify a value for this parameter unless IBM Software Support recommends a specific value.

This parameter does not have a default value.

An equivalent SYSIN keyword for this parameter does not exist.

In previous releases of DB2 HPU, the name of this parameter was VUU063/ULOPTLX.

Parameter to identify ROW-TRANSACTION-TIMESTAMP (VUU065/ULRTTST)

This parameter is optional. It specifies the technique to be used when generating the load SYSIN to identify the columns that are defined as ROW-TRANSACTION-TIMESTAMP.

This parameter applies only to natively processed SELECT statements.

Unsupported SELECT statements and SELECT statements that are processed with the DB2 FORCE option are ignored. In this case, the INGOREFIELDS and PERIODOVERRIDE keywords are not generated.

IGNORE

The load generates the column data. The generated load control statement contains the combination of IGNOREFIELDS keyword and a dummy field name for the identify column.

IGNOREOVERRIDE

The unloaded data is loaded into the identify column. The PERIODOVERRIDE keyword is added to the load control statement, and the exact name of the identify column is kept. This option is available for DB2 Version 9 and later releases. For earlier version of DB2, the ignore version is used.

IGNORE the default value.

Parameter to identify ROW-TRANSACTION-START-ID (VUU066/ULRTSID)

This parameter is optional. It specifies the technique to be used when generating the load SYSIN to identify the columns that are defined as ROW-TRANSACTION-START-ID.

This parameter applies only to natively processed SELECT statements.

Unsupported SELECT statements and SELECT statements that are processed with the DB2 FORCE option are ignored. In this case, the IGNOREFIELDS and PERIODOVERRIDE keywords are not generated.

IGNORE

The load generates the column data. The generated load control statement contains the combination of IGNOREFIELDS keyword and a dummy field name for the identify column.

TRANSIDOVERRIDE

The unloaded data is loaded into the identify column. The TRANSIDOVERRIDE keyword is added to the load control statement, and the exact name of the identify column is kept. This option is available for DB2 Version 9 and later releases. For earlier version of DB2, the ignore version is used.

IGNORE the default value.

Parameter to specify the default behavior regarding data truncation implied by CCSID conversion (VUU067-/ULCNVTRC)

This parameter is optional. It allows you to specify what policy to apply when data truncation is needed because of a CCSID conversion. The value of ULCNVTRC is overridden by the CONVERSION_TRUNCATION_ALLOWED sysin option (see Technical Parameters option block). Allowed values: YES/NO

YES Truncation of the output field after a CCSID conversion is allowed.

NO Truncation of the output field after a CCSID conversion is forbidden. Records requesting truncation are discarded and the return code is set to 4.

The default value is YES.

Related reference:

“User-allocated ddnames” on page 80

To run unload jobs, you must allocate certain ddnames in the DB2 HPU JCL.

DB2 HPU output data parameters

The DB2 HPU output data parameters section on the Product Parameters panel (CCQPPRD) in Tools Customizer contains the parameters for configuring output data.

You can accept the default values for the DB2 HPU output data parameters, or you can configure them based on your specific requirements.

The following list describes the DB2 HPU parameters. The parameters are listed in the following format:

description (Vxxxxnnn/parameter-name)

Override DCB DB2 HPU parameters (VUM029/DFSIGDCB)

This parameter is optional. It specifies whether the DCB JCL parameter can be overridden in DB2 HPU syntax.

Specify one of the following values:

YES For jobs that use DB2 HPU syntax, DB2 HPU ignores the DCB parameter in the JCL and sets the DCB attributes to the appropriate values.

NO For jobs that use DB2 HPU syntax, DB2 HPU uses the DCB parameter in the JCL.

The default value is NO.

The corresponding SYSIN keyword is DFSIGDCB in the Technical Parameters options block, which can be coded in the GLOBAL OPTIONS block.

In previous releases of DB2 HPU, the name of this parameter was VUM029/DFSIGDCB.

Override DCB UNLOAD PLUS syntax (VUM029/DFSIGDCB)

This parameter is optional. It specifies whether the DCB JCL parameter can be overridden in UNLOAD PLUS syntax.

Specify one of the following values:

YES For jobs that use UNLOAD PLUS syntax, DB2 HPU ignores the DCB parameter in the JCL and sets the DCB attributes to the appropriate values.

NO For jobs that use UNLOAD PLUS syntax, DB2 HPU uses the DCB parameter in the JCL.

The default value is NO.

The corresponding SYSIN keyword is DFSIGDCB in the Technical Parameters options block, which can be coded in the GLOBAL OPTIONS block.

In previous releases of DB2 HPU, the name of this parameter was VUM029/DFSIGDCB.

Override DCB FAST UNLOAD syntax (VUM029/DFSIGDCB)

This parameter is optional. It specifies whether the DCB JCL parameter can be overridden in Fast Unload syntax.

Specify one of the following values:

YES For jobs that use Fast Unload syntax, DB2 HPU ignores the DCB parameter in the JCL and sets the DCB attributes to the appropriate values.

NO For jobs that use Fast Unload syntax, DB2 HPU uses the DCB parameter in the JCL.

The default value is NO.

| The corresponding SYSIN keyword is DFSIGDCB in the Technical Parameters
| options block, which can be coded in the GLOBAL OPTIONS block.

| In previous releases of DB2 HPU, the name of this parameter was
| VUM029/DFSIGDCB.

| **NULL indicator, format USER (VUU014/ULNULL)**

| This parameter is optional. Use this parameter to modify the null or not-null
| indicator.

| **OFF**

| The null indicator is not present in the output data set.

| **hhhh**

| The first two digits (one hexadecimal character) represent the null indicator
| for a null column. The last two digits (one hexadecimal character)
| represent the null indicator for a not-null column.

| The default value is FF00.

| The corresponding SYSIN parameter is OPTIONS NULL.

| In previous releases of DB2 HPU, the name of this parameter was
| VUU014/ULNULL.

| **NULL indicator, format DSNTIAUL (VUU014/ULNULL)**

| This parameter is optional. Use this parameter to modify the null or not-null
| indicator.

| **OFF**

| The null indicator is not present in the output data set.

| **hhhh**

| The first two digits (one hexadecimal character) represent the null indicator
| for a null column. The last two digits (one hexadecimal character)
| represent the null indicator for a not-null column.

| The default value is FF00.

| The corresponding SYSIN parameter is OPTIONS NULL.

| In previous releases of DB2 HPU, the name of this parameter was
| VUU014/ULNULL.

| **NULL indicator, format VARIABLE (VUU014/ULNULL)**

| This parameter is optional. Use this parameter to modify the null or not-null
| indicator.

| **OFF**

| The null indicator is not present in the output data set.

| **hhhh**

| The first two digits (one hexadecimal character) represent the null indicator
| for a null column. The last two digits (one hexadecimal character)
| represent the null indicator for a not-null column.

| The default value is FF00.

| The corresponding SYSIN parameter is OPTIONS NULL.

| In previous releases of DB2 HPU, the name of this parameter was
| VUU014/ULNULL.

| **NULL indicator, format EXTERNAL (VUU014/ULNULL)**

| This parameter is optional. Use this parameter to modify the null or not-null
| indicator.

| **OFF**

| The null indicator is not present in the output data set.

| **hhhh**

| The first two digits (one hexadecimal character) represent the null indicator
| for a null column. The last two digits (one hexadecimal character)
| represent the null indicator for a not-null column.

| The default value is FF00.

| The corresponding SYSIN parameter is OPTIONS NULL.

| In previous releases of DB2 HPU, the name of this parameter was
| VUU014/ULNULL.

| **DATE, format USER (VUU015/ULDATE)**

| This parameter is optional. It specifies the default conversion type for a date
| column.

| The default value is DATE_C.

| The corresponding SYSIN parameter is OPTIONS DATE.

| In previous releases of DB2 HPU, the name of this parameter was
| VUU015/ULDATE.

| **DATE, format DSNTIAUL (VUU015/ULDATE)**

| This parameter is optional. It specifies the default conversion type for a date
| column.

| The default value is DATE_DB2.

| The corresponding SYSIN parameter is OPTIONS DATE.

| In previous releases of DB2 HPU, the name of this parameter was
| VUU015/ULDATE.

| **DATE, format DELIMITED (VUU015/ULDATE)**

| This parameter is optional. It specifies the default conversion type for a date
| column.

| The default value is DATE_C.

| The corresponding SYSIN parameter is OPTIONS DATE.

| In previous releases of DB2 HPU, the name of this parameter was
| VUU015/ULDATE.

| **DATE, format VARIABLE (VUU015/ULDATE)**

| This parameter is optional. It specifies the default conversion type for a date
| column.

| The default value is DATE_C.

| The corresponding SYSIN parameter is OPTIONS DATE.

| In previous releases of DB2 HPU, the name of this parameter was
| VUU015/ULDATE.

| **DATE, format EXTERNAL (VUU015/ULDATE)**

| This parameter is optional. It specifies the default conversion type for a date
| column.

The default value is DATE_DB2.

The corresponding SYSIN parameter is OPTIONS DATE.

In previous releases of DB2 HPU, the name of this parameter was VUU015/ULDATE.

TIME, format USER (VUU016/ULTIME)

This parameter is optional. It specifies the default conversion type for a time column.

The default value is TIME_A.

The corresponding SYSIN parameter is OPTIONS TIME.

In previous releases of DB2 HPU, the name of this parameter was VUU016/ULTIME.

TIME, format DSNTIAUL (VUU016/ULTIME)

This parameter is optional. It specifies the default conversion type for a time column.

The default value is TIME_DB2.

The corresponding SYSIN parameter is OPTIONS TIME.

In previous releases of DB2 HPU, the name of this parameter was VUU016/ULTIME.

TIME, format DELIMITED (VUU016/ULTIME)

This parameter is optional. It specifies the default conversion type for a time column.

The default value is TIME_A.

The corresponding SYSIN parameter is OPTIONS TIME.

In previous releases of DB2 HPU, the name of this parameter was VUU016/ULTIME.

TIME, format VARIABLE (VUU016/ULTIME)

This parameter is optional. It specifies the default conversion type for a time column.

The default value is TIME_A.

The corresponding SYSIN parameter is OPTIONS TIME.

In previous releases of DB2 HPU, the name of this parameter was VUU016/ULTIME.

TIME, format EXTERNAL (VUU016/ULTIME)

This parameter is optional. It specifies the default conversion type for a time column.

The default value is TIME_DB2.

The corresponding SYSIN parameter is OPTIONS TIME.

In previous releases of DB2 HPU, the name of this parameter was VUU016/ULTIME.

TIMESTAMP, format USER (VUU017/ULTMSTP)

This parameter is optional. It specifies the default conversion type for a timestamp column.

The default value is TMSTP_B

The corresponding SYSIN parameter is OPTIONS TIMESTAMP.

In previous releases of DB2 HPU, the name of this parameter was VUU017/ULTMSTP.

TIMESTAMP, format DSNTIAUL (VUU017/ULTMSTP)

This parameter is optional. It specifies the default conversion type for a timestamp column.

The default value is TMSTP_B

The corresponding SYSIN parameter is OPTIONS TIMESTAMP.

In previous releases of DB2 HPU, the name of this parameter was VUU017/ULTMSTP.

TIMESTAMP, format DELIMITED (VUU017/ULTMSTP)

This parameter is optional. It specifies the default conversion type for a timestamp column.

The default value is TMSTP_B

The corresponding SYSIN parameter is OPTIONS TIMESTAMP.

In previous releases of DB2 HPU, the name of this parameter was VUU017/ULTMSTP.

TIMESTAMP, format VARIABLE (VUU017/ULTMSTP)

This parameter is optional. It specifies the default conversion type for a timestamp column.

The default value is TMSTP_B

The corresponding SYSIN parameter is OPTIONS TIMESTAMP.

In previous releases of DB2 HPU, the name of this parameter was VUU017/ULTMSTP.

TIMESTAMP, format EXTERNAL (VUU017/ULTMSTP)

This parameter is optional. It specifies the default conversion type for a timestamp column.

The default value is TMSTP_B

The corresponding SYSIN parameter is OPTIONS TIMESTAMP.

In previous releases of DB2 HPU, the name of this parameter was VUU017/ULTMSTP.

Decimal picture, format USER (VUU018/ULPIC)

This parameter is optional. It specifies the numeric data display format.

Specify a value in the following format: *signpositionseparator*.

sign

Specify one of the following values to print the sign:

+ The plus sign (+) is used for positive values.

- The minus sign (-) is used for negative values.

P The padding character is used for positive values, and the minus sign (-) is used for negative values.

The default value for specifying the rules for printing is the minus sign.

position

Specify one of the following values to position the sign:

|
| **LEAD**

| The sign is placed in front of the numeric value. The LEAD value is
| ignored for floating point numbers.

| **TRAIL**

| The sign is placed after the numeric value. The TRAIL value is ignored
| for floating point numbers.

|
| The default value for where to position the sign is LEAD.

| **separator**

| Specify one of the following values for the decimal separator:

- | . Use a period (.) as the decimal separator.
- | , Use a comma (,) as the decimal separator.

|
| The default value -LEAD., which means that the sign is printed before the
| numeric value, the sign is shown only for negative values, and the period is
| used as the decimal separator.

| The corresponding SYSIN parameter is OPTIONS PIC.

| In previous releases of DB2 HPU, the name of this parameter was
| VUU018/ULPIC.

| **Decimal picture, format DSNTIAUL (VUU018/ULPIC)**

| This parameter is optional. It specifies the numeric data display format.

| Specify a value in the following format: *signpositionseparator*.

| **sign**

| Specify one of the following values to print the sign:

- | + The plus sign (+) is used for positive values.
- | - The minus sign (-) is used for negative values.
- | P The padding character is used for positive values, and the minus sign
| (-) is used for negative values.

|
| The default value for specifying the rules for printing is the minus sign.

| **position**

| Specify one of the following values to position the sign:

| **LEAD**

| The sign is placed in front of the numeric value. The LEAD value is
| ignored for floating point numbers.

| **TRAIL**

| The sign is placed after the numeric value. The TRAIL value is ignored
| for floating point numbers.

|
| The default value for where to position the sign is LEAD.

| **separator**

| Specify one of the following values for the decimal separator:

- | . Use a period (.) as the decimal separator.
- | , Use a comma (,) as the decimal separator.

The default value -LEAD., which means that the sign is printed before the numeric value, the sign is shown only for negative values, and the period is used as the decimal separator.

The corresponding SYSIN parameter is OPTIONS PIC.

In previous releases of DB2 HPU, the name of this parameter was VUU018/ULPIC.

Decimal picture, format DELIMITED (VUU018/ULPIC)

This parameter is optional. It specifies the numeric data display format.

Specify a value in the following format: *signpositionseparator*.

sign

Specify one of the following values to print the sign:

- + The plus sign (+) is used for positive values.
- The minus sign (-) is used for negative values.
- P The padding character is used for positive values, and the minus sign (-) is used for negative values.

The default value for specifying the rules for printing is the minus sign.

position

Specify one of the following values to position the sign:

LEAD

The sign is placed in front of the numeric value. The LEAD value is ignored for floating point numbers.

TRAIL

The sign is placed after the numeric value. The TRAIL value is ignored for floating point numbers.

The default value for where to position the sign is LEAD.

separator

Specify one of the following values for the decimal separator:

- . Use a period (.) as the decimal separator.
- , Use a comma (,) as the decimal separator.

The default value -LEAD., which means that the sign is printed before the numeric value, the sign is shown only for negative values, and the period is used as the decimal separator.

The corresponding SYSIN parameter is OPTIONS PIC.

In previous releases of DB2 HPU, the name of this parameter was VUU018/ULPIC.

Decimal picture, format VARIABLE (VUU018/ULPIC)

This parameter is optional. It specifies the numeric data display format.

Specify a value in the following format: *signpositionseparator*.

sign

Specify one of the following values to print the sign:

- + The plus sign (+) is used for positive values.
- The minus sign (-) is used for negative values.

P The padding character is used for positive values, and the minus sign (-) is used for negative values.

The default value for specifying the rules for printing is the minus sign.

position

Specify one of the following values to position the sign:

LEAD

The sign is placed in front of the numeric value. The LEAD value is ignored for floating point numbers.

TRAIL

The sign is placed after the numeric value. The TRAIL value is ignored for floating point numbers.

The default value for where to position the sign is LEAD.

separator

Specify one of the following values for the decimal separator:

. Use a period (.) as the decimal separator.

, Use a comma (,) as the decimal separator.

The default value -LEAD., which means that the sign is printed before the numeric value, the sign is shown only for negative values, and the period is used as the decimal separator.

The corresponding SYSIN parameter is OPTIONS PIC.

In previous releases of DB2 HPU, the name of this parameter was VUU018/ULPIC.

Decimal picture, format EXTERNAL (VUU018/ULPIC)

This parameter is optional. It specifies the numeric data display format.

Specify a value in the following format: *signpositionseparator*.

sign

Specify one of the following values to print the sign:

+ The plus sign (+) is used for positive values.

- The minus sign (-) is used for negative values.

P The padding character is used for positive values, and the minus sign (-) is used for negative values.

The default value for specifying the rules for printing is the minus sign.

position

Specify one of the following values to position the sign:

LEAD

The sign is placed in front of the numeric value. The LEAD value is ignored for floating point numbers.

TRAIL

The sign is placed after the numeric value. The TRAIL value is ignored for floating point numbers.

The default value for where to position the sign is LEAD.

separator

Specify one of the following values for the decimal separator:

- Use a period (.) as the decimal separator.
- Use a comma (,) as the decimal separator.

The default value -LEAD., which means that the sign is printed before the numeric value, the sign is shown only for negative values, and the period is used as the decimal separator.

The corresponding SYSIN parameter is OPTIONS PIC.

In previous releases of DB2 HPU, the name of this parameter was VUU018/ULPIC.

Display format for numeric values, format USER (VUU032/ULMASK)

This parameter is optional. It specifies the display format for numeric values (zero and separator).

The value of this parameter consists of four separate columns:

- The Left padding column indicates whether the number is padded on the left with nonsignificant zeroes to fill up the output field. The sign character is added to the left of these zeroes.
- The Left zero column indicates whether a zero is placed to the left of the decimal separator when the value is 0.
- The Decimal separator column indicates whether the decimal separator is displayed. The Decimal separator column applies only to decimal data. The decimal separator is never displayed for SMALLINT or INTEGER values.
- The Right padding column indicates whether the number is padded on the right of the decimal separator with zeroes, up to the number of digits of the decimal scale.

Important:

1. The ULMASK variable has no impact on the formatting of FLOAT values.

Table 30. Nonsignificant zero values for DECIMAL and SMALLINT/INTEGER

ULMASK value	Left padding	Left zero	Decimal separator	Right padding
*.	No	Only when the value is 0	Only if the decimal value is not 0	No
0.*	No	Always	Only if the decimal value is not 0	No
*.0	No	Only when the decimal scale is 0	Only if the decimal scale is not 0	Yes
0.0	No	Always	Only if the decimal scale is not 0	Yes
00.0	Yes	Always	Only if the decimal scale is not 0	Yes
*.Z	No	Only when the decimal scale is 0	Always	Yes

Table 30. Nonsignificant zero values for DECIMAL and SMALLINT/INTEGER (continued)

ULMASK value	Left padding	Left zero	Decimal separator	Right padding
0.Z	No	Always	Always	Yes
00.Z	Yes	Always	Always	Yes

For examples of the effect of using ULMASK, see “OPTIONS block syntax and description” on page 134.

The following example shows how to specify a ULMASK value for the USER format:

```
00.0
```

The default value when a LIKE value is used to force conversion to a CHAR, an INTO clause, or a REFORMAT clause is *.*

The corresponding SYSIN parameter is OPTIONS PIC.

In previous releases of DB2 HPU, the name of this parameter was VUU032/ULMASK.

Display format for numeric values, format DSNTIAUL (VUU032/ULMASK)

This parameter is optional. It specifies the display format for numeric values (zero and separator).

The value of this parameter consists of four separate columns:

- The Left padding column indicates whether the number is padded on the left with nonsignificant zeroes to fill up the output field. The sign character is added to the left of these zeroes.
- The Left zero column indicates whether a zero is placed to the left of the decimal separator when the value is 0.
- The Decimal separator column indicates whether the decimal separator is displayed. The Decimal separator column applies only to decimal data. The decimal separator is never displayed for SMALLINT or INTEGER values.
- The Right padding column indicates whether the number is padded on the right of the decimal separator with zeroes, up to the number of digits of the decimal scale.

Important:

1. The ULMASK variable has no impact on the formatting of FLOAT values.

Table 31. Nonsignificant zero values for DECIMAL and SMALLINT/INTEGER

ULMASK value	Left padding	Left zero	Decimal separator	Right padding
.	No	Only when the value is 0	Only if the decimal value is not 0	No
0.*	No	Always	Only if the decimal value is not 0	No
*.0	No	Only when the decimal scale is 0	Only if the decimal scale is not 0	Yes

Table 31. Nonsignificant zero values for DECIMAL and SMALLINT/INTEGER (continued)

ULMASK value	Left padding	Left zero	Decimal separator	Right padding
0.0	No	Always	Only if the decimal scale is not 0	Yes
00.0	Yes	Always	Only if the decimal scale is not 0	Yes
*.Z	No	Only when the decimal scale is 0	Always	Yes
0.Z	No	Always	Always	Yes
00.Z	Yes	Always	Always	Yes

For examples of the effect of using ULMASK, see “OPTIONS block syntax and description” on page 134.

The following example shows how to specify a ULMASK value for the USER format:

```
00.0
```

The default value when a LIKE value is used to force conversion to a CHAR, an INTO clause, or a REFORMAT clause is *.*

The corresponding SYSIN parameter is OPTIONS PIC.

In previous releases of DB2 HPU, the name of this parameter was VUU032/ULMASK.

Display format for numeric values, format DELIMITED (VUU032/ULMASK)

This parameter is optional. It specifies the display format for numeric values (zero and separator).

The value of this parameter consists of four separate columns:

- The Left padding column indicates whether the number is padded on the left with nonsignificant zeroes to fill up the output field. The sign character is added to the left of these zeroes.
- The Left zero column indicates whether a zero is placed to the left of the decimal separator when the value is 0.
- The Decimal separator column indicates whether the decimal separator is displayed. The Decimal separator column applies only to decimal data. The decimal separator is never displayed for SMALLINT or INTEGER values.
- The Right padding column indicates whether the number is padded on the right of the decimal separator with zeroes, up to the number of digits of the decimal scale.

Important:

1. The ULMASK variable has no impact on the formatting of FLOAT values.

Table 32. Nonsignificant zero values for DECIMAL and SMALLINT/INTEGER

ULMASK value	Left padding	Left zero	Decimal separator	Right padding
.	No	Only when the value is 0	Only if the decimal value is not 0	No

Table 32. Nonsignificant zero values for DECIMAL and SMALLINT/INTEGER (continued)

ULMASK value	Left padding	Left zero	Decimal separator	Right padding
0.*	No	Always	Only if the decimal value is not 0	No
*.0	No	Only when the decimal scale is 0	Only if the decimal scale is not 0	Yes
0.0	No	Always	Only if the decimal scale is not 0	Yes
00.0	Yes	Always	Only if the decimal scale is not 0	Yes
*.Z	No	Only when the decimal scale is 0	Always	Yes
0.Z	No	Always	Always	Yes
00.Z	Yes	Always	Always	Yes

For examples of the effect of using ULMASK, see “OPTIONS block syntax and description” on page 134.

The following example shows how to specify a ULMASK value for the USER format:

```
00.0
```

The default value when a LIKE value is used to force conversion to a CHAR, an INTO clause, or a REFORMAT clause is *.*

The corresponding SYSIN parameter is OPTIONS PIC.

In previous releases of DB2 HPU, the name of this parameter was VUU032/ULMASK.

Display format for numeric values, format VARIABLE

This parameter is optional. It specifies the display format for numeric values (zero and separator).

The value of this parameter consists of four separate columns:

- The Left padding column indicates whether the number is padded on the left with nonsignificant zeroes to fill up the output field. The sign character is added to the left of these zeroes.
- The Left zero column indicates whether a zero is placed to the left of the decimal separator when the value is 0.
- The Decimal separator column indicates whether the decimal separator is displayed. The Decimal separator column applies only to decimal data. The decimal separator is never displayed for SMALLINT or INTEGER values.
- The Right padding column indicates whether the number is padded on the right of the decimal separator with zeroes, up to the number of digits of the decimal scale.

Important:

1. The ULMASK variable has no impact on the formatting of FLOAT values.

Table 33. Nonsignificant zero values for DECIMAL and SMALLINT/INTEGER

ULMASK value	Left padding	Left zero	Decimal separator	Right padding
.*	No	Only when the value is 0	Only if the decimal value is not 0	No
0.*	No	Always	Only if the decimal value is not 0	No
*.0	No	Only when the decimal scale is 0	Only if the decimal scale is not 0	Yes
0.0	No	Always	Only if the decimal scale is not 0	Yes
00.0	Yes	Always	Only if the decimal scale is not 0	Yes
*.Z	No	Only when the decimal scale is 0	Always	Yes
0.Z	No	Always	Always	Yes
00.Z	Yes	Always	Always	Yes

For examples of the effect of using ULMASK, see “OPTIONS block syntax and description” on page 134.

The following example shows how to specify a ULMASK value for the USER format:

```
00.0
```

The default value when a LIKE value is used to force conversion to a CHAR, an INTO clause, or a REFORMAT clause is *.*

The corresponding SYSIN parameter is OPTIONS PIC.

In previous releases of DB2 HPU, the name of this parameter was VUU032/ULMASK.

Display format for numeric values, format EXTERNAL (VUU032/ULMASK)

This parameter is optional. It specifies the display format for numeric values (zero and separator).

The value of this parameter consists of four separate columns:

- The Left padding column indicates whether the number is padded on the left with nonsignificant zeroes to fill up the output field. The sign character is added to the left of these zeroes.
- The Left zero column indicates whether a zero is placed to the left of the decimal separator when the value is 0.
- The Decimal separator column indicates whether the decimal separator is displayed. The Decimal separator column applies only to decimal data. The decimal separator is never displayed for SMALLINT or INTEGER values.
- The Right padding column indicates whether the number is padded on the right of the decimal separator with zeroes, up to the number of digits of the decimal scale.

Important:

1. The ULMASK variable has no impact on the formatting of FLOAT values.

Table 34. Nonsignificant zero values for DECIMAL and SMALLINT/INTEGER

ULMASK value	Left padding	Left zero	Decimal separator	Right padding
,	No	Only when the value is 0	Only if the decimal value is not 0	No
0.*	No	Always	Only if the decimal value is not 0	No
*.0	No	Only when the decimal scale is 0	Only if the decimal scale is not 0	Yes
0.0	No	Always	Only if the decimal scale is not 0	Yes
00.0	Yes	Always	Only if the decimal scale is not 0	Yes
*.Z	No	Only when the decimal scale is 0	Always	Yes
0.Z	No	Always	Always	Yes
00.Z	Yes	Always	Always	Yes

For examples of the effect of using ULMASK, see “OPTIONS block syntax and description” on page 134.

The following example shows how to specify a ULMASK value for the USER format:

```
00.0
```

The default value when a LIKE value is used to force conversion to a CHAR, an INTO clause, or a REFORMAT clause is *.*

The corresponding SYSIN parameter is OPTIONS PIC.

In previous releases of DB2 HPU, the name of this parameter was VUU032/ULMASK.

Options for global LOAD statement (VUU019/ULOPTLDT)

This parameter is optional. It defines the parameters of the LOAD statement that were generated at the table space level.

Enclose all sub-parameters in parentheses, as shown in the following example:

```
SORTDEVT(SYSDA) SORTNUM(32)
```

To generate a parenthesis in the LOADDDN, include two parentheses in this parameter, as shown in the following statement:

```
(ENFORCE((NO)),LOG(NO),COPYDDN((SYSCOPY)))
```

This statement generates this option in the LOAD statement:

```
ENFORCE(NO) LOG NO COPYDDN(SYSCOPY)
```

To code the load options on several lines, follow these rules:

- Use a continuation comma at the end of each line.
- Begin the first string with a left parenthesis, and code the corresponding right parenthesis only on the last line used.

The following parameters are valid. If you code a parameter that is not in this list, it is not controlled, and it is copied in the generated LOAD SYSIN.

- COPYDDN
- DISCARDN
- DISCARDS
- ENFORCE(CONSTRAINTS/NO)
- INDDN
- KEEPDICTIONARY
- LOG(YES/NO)
- RECOVERYDDN
- REPLACE
- RESUME(YES|NO)
- SORTDEVT
- SORTKEYS
- SORTNUM
- PARALLEL or PARALLEL (num-subtask) where num-subtask specifies the maximum number of sub-tasks that are to be processed in parallel by the LOAD utility. The value must be an integer between 0 and 32767, inclusive.

Important: If you specify the SORTKEYS keyword, the value that you specify is substituted with a value that is calculated according to the number of unloaded records, except when you run DB2 HPU in EXECUTE NO mode. When you run DB2 HPU in EXECUTE NO mode, the &SORTKEYS variable is replaced with 0.

If you code a parameter that is not in the previous list, it will not be controlled, and will only be copied in the generated LOAD SYSIN.

When the output file is a spanned file, NUMRECS is always generated in the SYSIN for the LOAD utility, therefore the SORTKEYS parameter is not taken into account.

The default value is (LOG (NO), ENFORCE(NO)).

In previous releases of DB2 HPU, the name of this parameter was VUU019/ULOPTLDT.

Options of partition LOAD statement (VUU020/ULOPTLDP)

This parameter is optional. Defines the parameters of the LOAD statement that was generated at the partition level.

To code the load options on several lines, follow these rules:

- Use a continuation comma at the end of each line.
- Begin the first string with a left parenthesis, and code the corresponding right parenthesis only on the last line used.

The following parameters are accepted:

- RESUME(YES|NO)
- REPLACE
- KEEPDICTIONARY

The default value is (RESUME(YES)).

In previous releases of DB2 HPU, the name of this parameter was VUU020/ULOPTLDP.

Options of the table LOAD statement (VUU070/ULOPTLDP)

This parameter is optional. Defines the parameters of the LOAD statement to be generated at the table level.

To code the load options on several lines, follow these rules:

- Use a continuation comma at the end of each line.
- Begin the first string with a left parenthesis, and code the corresponding right parenthesis only on the last line used.

The following parameters are accepted:

- NUMREC(&NUMREC): the number of unloaded records is specified as the operand of the NUMRECS options written in the generated LOAD statement
- NUMRECS(integer): the specified integer value is used as the operand of the NUMRECS options written in the generated LOAD statement.

If you code a parameter that is not in the above, it will not be controlled and will only be copied in the generated LOAD SYSIN as specified.

The default value is (NUMRECS(&NUMRECS)).

In previous releases of DB2 HPU, the name of this parameter was VUU020/ULOPTLDP.

Position for NULL indicator, format USER (VUU022/NULLPOS)

This parameter is optional. It specifies the position of the NULL indicator within the DB2 HPU output data sets.

Specify one of the following values:

AFTER

The NULL indicator will be set after the column data.

BEFORE

The NULL indicator will be set before the column data.

The default value is BEFORE.

The corresponding SYSIN parameter is OPTIONS NULLPOS.

In previous releases of DB2 HPU, the name of this parameter was VUU022/NULLPOS.

Position for NULL indicator, format DSNTIAUL (VUU022/NULLPOS)

This parameter is optional. It specifies the position of the NULL indicator within the DB2 HPU output data sets.

Specify one of the following values:

AFTER

The NULL indicator will be set after the column data.

BEFORE

The NULL indicator will be set before the column data.

The default value is BEFORE.

The corresponding SYSIN parameter is OPTIONS NULLPOS.

In previous releases of DB2 HPU, the name of this parameter was VUU022/NULLPOS.

| **Position for NULL indicator, format VARIABLE (VUU022/NULLPOS)**

| This parameter is optional. It specifies the position of the NULL indicator
| within the DB2 HPU output data sets.

| Specify one of the following values:

| **AFTER**

| The NULL indicator will be set after the column data.

| **BEFORE**

| The NULL indicator will be set before the column data.

| The default value is BEFORE.

| The corresponding SYSIN parameter is OPTIONS NULLPOS.

| In previous releases of DB2 HPU, the name of this parameter was
| VUU022/NULLPOS.

| **Position for NULL indicator, format EXTERNAL (VUU022/NULLPOS)**

| This parameter is optional. It specifies the position of the NULL indicator
| within the DB2 HPU output data sets.

| Specify one of the following values:

| **AFTER**

| The NULL indicator will be set after the column data.

| **BEFORE**

| The NULL indicator will be set before the column data.

| The default value is BEFORE.

| The corresponding SYSIN parameter is OPTIONS NULLPOS.

| In previous releases of DB2 HPU, the name of this parameter was
| VUU022/NULLPOS.

| **Use the same DDN for UNLDDN, OUTDDN, and LOADDN (VUU029/UNLDDN)**

| This parameter is optional. Use this parameter to prevent the use of the same
| DDN for UNLDDN, OUTDDN, and LOADDN.

| Specify one of the following values:

| **YES**

| If you use the same DDN in a single SELECT statement with the keywords
| UNLDDN, OUTDDN, and LOADDN, DB2 HPU issues error INZU124E.

| **NO** An error is not issued when you use the same DDN in a single SELECT
| statement with the keywords UNLDDN, OUTDDN, and LOADDN.

| The default value is NO.

| In previous releases of DB2 HPU, the name of this parameter was
| VUU029/UNLDDN.

| **DATE/TIME delimiter option (VUU031/DTDELIM)**

| This parameter is optional. It applies to FORMAT DELIMITED to delimit the
| column types DATE, TIME, and TIMESTAMP in the output data set.

| Specify one of the following values:

| **YES**

| Column types DATE, TIME, and TIMESTAMP are delimited by the column
| delimiter that is specified by the option DELIM *literal*.

NO If the NULL DELIM option is used in FORMAT DELIMITED, this parameter will not apply to the DATE, TIME, and TIMESTAMP columns.

The default value is NO.

In previous releases of DB2 HPU, the name of this parameter was VUU031/DTDELIM.

NULL DATE/TIME delimiter option (VUU043/DTNULDLM)

This parameter is optional. Specifies that the NULL DELIM option of FORMAT DELIMITED should also be used for DATE, TIME, and TIMESTAMP columns. This parameter is used only when PARMLIB variable VUU031/DTDELIM is set to YES.

YES

If the NULL DELIM option is used in FORMAT DELIMITED, it will also apply to the DATE, TIME, and TIMESTAMP columns.

NO If the NULL DELIM option is used in FORMAT DELIMITED, it will not apply to the DATE, TIME, and TIMESTAMP columns.

The default value is NO.

In previous releases of DB2 HPU, the name of this parameter was VUU043/DTNULDLM.

Technique to generate SYSIN for identity column (VUU058/ULIDENT)

This parameter is optional. It specifies the technique to be used when generating the load SYSIN for identity columns that are defined as GENERATED ALWAYS.

This parameter applies only to natively processed SELECT statements.

Unsupported SELECT statements and SELECT statements that are processed with the DB2 FORCE option are ignored. In this case, the IGNOREFIELDS and IDENTITYOVERRIDE keywords are not generated.

Specify one of the following values:

IGNORE

The load generates the column data. The generated load control statement contains the combination of IGNOREFIELDS keyword and a dummy field name for the identity column.

OVERRIDE

The unloaded data is loaded into the identity column. The IDENTITYOVERRIDE keyword is added to the load control statement, and the exact name of the identity column is kept. This option is available for DB2 Version 9 and later releases. For earlier versions of DB2, the IGNORE option is used.

The default value is IGNORE.

In previous releases of DB2 HPU, the name of this parameter was VUU058/ULIDENT.

Positive sign for zoned-decimal numeric values (VUU037/ULSIGZ)

This parameter is optional. It specifies the positive sign for decimal zoned values in hexadecimal format. Valid values are A, B, C, D, E, and F. Each value represents the positive sign. Using this parameter helps you create a customized format that meets the requirements of the programs that process the unloaded data.

The default value is C.

In previous releases of DB2 HPU, the name of this parameter was VUU037/ULSIGZ.

Positive sign for date-time packed values (VUU038/ULSIGDTP)

This parameter is optional. It specifies the positive sign for date- and time-packed values. Valid values are A, B, C, D, E, and F. Each value represents the positive sign. Using this parameter helps you create a customized format that meets the requirements of the programs that process the unloaded data.

The default value is F.

In previous releases of DB2 HPU, the name of this parameter was VUU038/ULSIGDTP.

Truncate variable records exceeding 32756 (VUU040/ULTR32K)

This parameter is optional. Specifies whether DB2 HPU will truncate the variable records that exceed the physical limit of 32756.

Specify one of the following values:

YES

The output file record format is variable, and the LRECL is greater than 32756 (LRECL > 32756). LRECL is truncated to 32756, and message INZU297I is issued.

NO The output file record format is fixed, and the LRECL is greater than 32756 (LRECL > 32756). Processing ends, and message INZU298E is issued.

The default value is NO.

In previous releases of DB2 HPU, the name of this parameter was VUU040/ULTR32K.

Default output format for a logical unload (VUU045/ULFORMAT)

This parameter is optional. It specifies the value of the output format when the FORMAT parameter is not specified in the SYSIN for a SELECT statement.

This parameter applies only to DB2 HPU syntax.

Attention: Changing this value in the PARMLIB will affect existing production jobs that do not specify the FORMAT parameter. Use caution when changing the value of this parameter in the PARMLIB.

Specify one of the following values:

DSNTIAUL

Specifies that the default output format for SELECT statements is FORMAT VARIABLE. This format is the default value.

STRICT

Use this value to unload data in the same format that the DSNTIAUL program produces. DSNTIAUL STRICT affects the formatting of constant character strings that are specified in SELECT statements.

Specifying STRICT in VUU045/ULFORMAT overrides the value of DSNTIAUL STRICT in the PARMLIB variable ULOPTNS. If STRICT is not specified in VUU045/ULFORMAT, the content of the PARMLIB variable ULOPTNS is used.

Important: FORMAT DSNTIAUL STRICT applies only to DB2 HPU syntax.

DELIMITED

Specifies that the default output format for SELECT statements is FORMAT DELIMITED.

SEP=*val* | BLANK

Specifies the separator character, *val*, to separate fields in the output data set. *val* can be specified in character ('c') or hexadecimal (X'hh') format. Values that are specified in hexadecimal are not converted in CCSID conversion. Values that are specified in character are converted, if necessary, into the output CCSID.

Use BLANK to specify a space character. BLANK is the default value for SEP.

DELIM=*val* | BLANK | NONE

Specifies the delimiter character, *val*, to be used to enclose CHAR, VARCHAR, GRAPHIC, and VARGRAPHIC fields in the output data set. *val* can be specified in character ('c') or hexadecimal (X'hh') format. Values that are specified in hexadecimal are not converted in CCSID conversion. Values that are specified in character are converted, if necessary, into the output CCSID.

BLANK

Specifies a space character.

NONE

Specifies that you do not want to use a delimiter.

The default value for the delimiter is NONE.

NULLDELIM

Specifies that null values are not enclosed by the delimiter character that is specified by the *val* of DELIM.

The DELIM and NULL DELIM keyword can also apply to DATE, TIME, TIMESTAMP format depending on the settings of VUU043/DTNULDLM.>

VARIABLE

Specifies that the default output format for SELECT statements is FORMAT VARIABLE.

ALL

If the column that was selected last is variable, the output data set is VB, and this last column is written on its effective length. The effective length is the actual length of the data that is contained in a column that has a variable type. Both length bytes precede the column.

END

All the variable columns are written by using their actual length.

The default value is END if only VARIABLE is specified.

EXTERNAL

Specifies that the default output format for SELECT statements is FORMAT EXTERNAL.

|
| **USER**

| Specifies that the default output format for SELECT statements is FORMAT
| USER.

|
| In previous releases of DB2 HPU, the name of this parameter was
| VUU045/ULFORMAT.

| **Size of the autotag in the output file (VUU046/ULAUTAG)**

| This parameter is optional. Use this parameter to specify a number for the
| value of the autotag that is generated in the output file. It applies only to DB2
| HPU syntax, and it applies to all output formats.

| Valid values are 0 - 8.

| The default value is 0. When the default value is specified, an autotag is not
| generated.

| The corresponding SYSIN parameter is OPTIONS AUTOTAG.

| **Attention:** Changing this value in the PARMLIB will affect existing
| production jobs. Use caution when you change this value.

| In previous releases of DB2 HPU, the name of this parameter was
| VUU046/ULAUTAG.

| **Full compatibility of LOAD statement with DB2 LOAD (VUU054/CTRLLIBM)**

| This parameter is optional. It specifies whether the LOAD statement that is
| generated by DB2 HPU must be fully compatible with the DB2 LOAD utility.

| Valid values are YES and NO.

| When this parameter is set to YES, additional controls are made when the
| DELIMITED format is requested, and the LOAD statement is generated only if
| it is compatible with the DB2 LOAD utility.

| In previous releases of DB2 HPU, the name of this parameter was
| VUU054/CTRLLIBM.

| **Delimiter for external date, format USER (VU055/DATEDEL)**

| This parameter is optional. It specifies the default delimiter that is used in
| external date representations.

| Specify one of the following values:

| 'c'

| The specified value must be in character format, coded between single
| quotation marks, and one byte long.

| **BLANK**

| Specifies that a space character is used as the default delimiter for external
| date representation. By default, this variable applies to FORMAT USER. It
| can be used to change the default value of the delimiter for date external
| representation for any format by using the syntax for defining formatting
| options for each output format.

| The default value is ' '.

| The corresponding SYSIN parameter is OPTIONS DATEDELIM.

| In previous releases of DB2 HPU, the name of this parameter was
| VUU055/DATEDEL.

| **Delimiter for external date, format DSNTIAUL (VU055/DATEDEL)**

| This parameter is optional. It specifies the default delimiter that is used in
| external date representations.

| Specify one of the following values:

| **'c'**

| The specified value must be in character format, coded between single
| quotation marks, and one byte long.

| **BLANK**

| Specifies that a space character is used as the default delimiter for external
| date representation. By default, this variable applies to FORMAT USER. It
| can be used to change the default value of the delimiter for date external
| representation for any format by using the syntax for defining formatting
| options for each output format.

| The default value is '-'.
|

| The corresponding SYSIN parameter is OPTIONS DATEDELIM.

| In previous releases of DB2 HPU, the name of this parameter was
| VUU055/DATEDEL.

| **Delimiter for external date, format DELIMITED (VU055/DATEDEL)**

| This parameter is optional. It specifies the default delimiter that is used in
| external date representations.

| Specify one of the following values:

| **'c'**

| The specified value must be in character format, coded between single
| quotation marks, and one byte long.

| **BLANK**

| Specifies that a space character is used as the default delimiter for external
| date representation. By default, this variable applies to FORMAT USER. It
| can be used to change the default value of the delimiter for date external
| representation for any format by using the syntax for defining formatting
| options for each output format.

| The default value is '-'.
|

| The corresponding SYSIN parameter is OPTIONS DATEDELIM.

| In previous releases of DB2 HPU, the name of this parameter was
| VUU055/DATEDEL.

| **Delimiter for external date, format VARIABLE (VU055/DATEDEL)**

| This parameter is optional. It specifies the default delimiter that is used in
| external date representations.

| Specify one of the following values:

| **'c'**

| The specified value must be in character format, coded between single
| quotation marks, and one byte long.

| **BLANK**

| Specifies that a space character is used as the default delimiter for external
| date representation. By default, this variable applies to FORMAT USER. It

can be used to change the default value of the delimiter for date external representation for any format by using the syntax for defining formatting options for each output format.

The default value is '-'.

The corresponding SYSIN parameter is OPTIONS DATEDELIM.

In previous releases of DB2 HPU, the name of this parameter was VUU055/DATEDEL.

Delimiter for external date, format EXTERNAL (VU055/DATEDEL)

This parameter is optional. It specifies the default delimiter that is used in external date representations.

Specify one of the following values:

'c'

The specified value must be in character format, coded between single quotation marks, and one byte long.

BLANK

Specifies that a space character is used as the default delimiter for external date representation. By default, this variable applies to FORMAT USER. It can be used to change the default value of the delimiter for date external representation for any format by using the syntax for defining formatting options for each output format.

The default value is '-'.

The corresponding SYSIN parameter is OPTIONS DATEDELIM.

In previous releases of DB2 HPU, the name of this parameter was VUU055/DATEDEL.

Delimiter in external time, format USER (VUU056/TIMEDEL)

This parameter is optional. It specifies the default delimiter that will be used in external time representations.

'c'

The specified value must be in character format, coded between single quotation marks, and one byte long.

BLANK

Specifies that a space character will be used as the default delimiter for external time representation.

The default value is '.'.

The corresponding SYSIN parameter is OPTIONS TIMEDELIM.

In previous releases of DB2 HPU, the name of this parameter was VUU056/TIMEDEL.

Delimiter in external time, format DSNTIAUL (VUU056/TIMEDEL)

This parameter is optional. It specifies the default delimiter that will be used in external time representations.

'c'

The specified value must be in character format, coded between single quotation marks, and one byte long.

| **BLANK**

| Specifies that a space character will be used as the default delimiter for
| external time representation.

| The default value is ' '.

| The corresponding SYSIN parameter is OPTIONS TIMEDELIM.

| In previous releases of DB2 HPU, the name of this parameter was
| VUU056/TIMEDEL.

| **Delimiter in external time, format DELIMITED (VUU056/TIMEDEL)**

| This parameter is optional. It specifies the default delimiter that will be used in
| external time representations.

| **'c'**

| The specified value must be in character format, coded between single
| quotation marks, and one byte long.

| **BLANK**

| Specifies that a space character will be used as the default delimiter for
| external time representation.

| The default value is ' '.

| The corresponding SYSIN parameter is OPTIONS TIMEDELIM.

| In previous releases of DB2 HPU, the name of this parameter was
| VUU056/TIMEDEL.

| **Delimiter in external time, format VARIABLE (VUU056/TIMEDEL)**

| This parameter is optional. It specifies the default delimiter that will be used in
| external time representations.

| **'c'**

| The specified value must be in character format, coded between single
| quotation marks, and one byte long.

| **BLANK**

| Specifies that a space character will be used as the default delimiter for
| external time representation.

| The default value is ' '.

| The corresponding SYSIN parameter is OPTIONS TIMEDELIM.

| In previous releases of DB2 HPU, the name of this parameter was
| VUU056/TIMEDEL.

| **Delimiter in external time, format EXTERNAL (VUU056/TIMEDEL)**

| This parameter is optional. It specifies the default delimiter that will be used in
| external time representations.

| **'c'**

| The specified value must be in character format, coded between single
| quotation marks, and one byte long.

| **BLANK**

| Specifies that a space character will be used as the default delimiter for
| external time representation.

| The default value is ' '.

| The corresponding SYSIN parameter is OPTIONS TIMEDELIM.

In previous releases of DB2 HPU, the name of this parameter was VUU056/TIMEDEL.

Options apply to all formats (VUU057/OPALLFMT)

This parameter is optional. It specifies whether the formatting options that are specified in SYSIN in the OPTIONS block at the GLOBAL level or the UNLOAD level apply only to FORMAT USER or to all output formats. VUU057/OPALLFMT applies to the following SYSIN parameters of the OPTIONS block:

- NULL
- DATE
- TIME
- TIMESTAMP
- PIC
- DATEDELIM
- TIMEDELIM
- NULLID
- NULLPOS
- NULLPAD
- PADDING
- TRIM

Specify one of the following values:

NO The SYSIN parameters in the previous list apply only to FORMAT USER when they are specified at the GLOBAL level or at the UNLOAD level.

YES

The SYSIN parameters in the previous list apply to all output formats when they are specified at the GLOBAL level or at the UNLOAD level.

Attention: Specifying YES for this PARMLIB variable might affect the output data that is generated by existing production jobs. Change this variable to YES only after you have verified that existing DB2 HPU production jobs will not be affected by the change.

When the SYSIN parameters in the previous list are specified at the SELECT level (after the OUTDDN keyword), they always apply to all output formats. In this case, the value of OPALLFMT is ignored.

The default value is NO.

In previous releases of DB2 HPU, the name of this parameter was VUU057/OPALLFMT.

Trim the trailing blanks, format USER (VUU059/ULTRIM)

This parameter is optional. Use it to specify whether DB2 HPU will remove the trailing blanks for character data that is unloaded into a variable length field. It applies to CHAR, VARCHAR, GRAPHIC, VARGRAPHIC, CLOB, and DBCLOB output fields. However, it does not apply to numeric external data.

Specify one of the following values:

NO Trailing blanks are not removed from variable length strings.

YES

Trailing blanks are removed when data is unloaded into VARCHAR, VARGRAPHIC output fields, or CLOB, and DBCLOB output field except if

SPANNED YES is requested. The length of the output field is adjusted to match the effective number of characters that are written.

The default value is NO.

In previous releases of DB2 HPU, the name of this parameter was VUU059/ULTRIM.

Trim the trailing blanks, format DSNTIAUL (VUU059/ULTRIM)

This parameter is optional. Use it to specify whether DB2 HPU will remove the trailing blanks for character data that is unloaded into a variable length field. It applies to CHAR, VARCHAR, GRAPHIC, VARGRAPHIC, CLOB, and DBCLOB output fields. However, it does not apply to numeric external data.

Specify one of the following values:

NO Trailing blanks are not removed from variable length strings.

YES

Trailing blanks are removed when data is unloaded into VARCHAR, VARGRAPHIC output fields, or CLOB, and DBCLOB output field except if SPANNED YES is requested. The length of the output field is adjusted to match the effective number of characters that are written.

The default value is NO.

In previous releases of DB2 HPU, the name of this parameter was VUU059/ULTRIM.

Trim the trailing blanks, format DELIMITED (VUU059/ULTRIM)

This parameter is optional. Use it to specify whether DB2 HPU will remove the trailing blanks for character data that is unloaded into a variable length field. It applies to CHAR, VARCHAR, GRAPHIC, VARGRAPHIC, CLOB, and DBCLOB output fields. However, it does not apply to numeric external data.

Specify one of the following values:

NO Trailing blanks are not removed from variable length strings.

YES

Trailing blanks are removed when data is unloaded into VARCHAR, VARGRAPHIC output fields, or CLOB, and DBCLOB output field except if SPANNED YES is requested. The length of the output field is adjusted to match the effective number of characters that are written.

The default value is NO.

In previous releases of DB2 HPU, the name of this parameter was VUU059/ULTRIM.

Trim the trailing blanks, format VARIABLE (VUU059/ULTRIM)

This parameter is optional. Use it to specify whether DB2 HPU will remove the trailing blanks for character data that is unloaded into a variable length field. It applies to CHAR, VARCHAR, GRAPHIC, VARGRAPHIC, CLOB, and DBCLOB output fields. However, it does not apply to numeric external data.

Specify one of the following values:

NO Trailing blanks are not removed from variable length strings.

YES

Trailing blanks are removed when data is unloaded into VARCHAR, VARGRAPHIC output fields, or CLOB, and DBCLOB output field except if

SPANNED YES is requested. The length of the output field is adjusted to match the effective number of characters that are written.

The default value is NO.

In previous releases of DB2 HPU, the name of this parameter was VUU059/ULTRIM.

Trim the trailing blanks, format EXTERNAL (VUU059/ULTRIM)

This parameter is optional. Use it to specify whether DB2 HPU will remove the trailing blanks for character data that is unloaded into a variable length field. It applies to CHAR, VARCHAR, GRAPHIC, VARGRAPHIC, CLOB, and DBCLOB output fields. However, it does not apply to numeric external data.

Specify one of the following values:

NO Trailing blanks are not removed from variable length strings.

YES

Trailing blanks are removed when data is unloaded into VARCHAR, VARGRAPHIC output fields, or CLOB, and DBCLOB output field except if SPANNED YES is requested. The length of the output field is adjusted to match the effective number of characters that are written.

The default value is NO.

In previous releases of DB2 HPU, the name of this parameter was VUU059/ULTRIM.

String constant is unloaded as CHAR/VARCHAR data types (VUU064/STRNGCST)

This parameter is optional. It specifies whether string constants in the SELECT statements are handled as CHAR or VARCHAR data when the SELECT statement is processed in native mode.

CHAR

The string constants are handled as CHAR data unless the DSNTIAUL STRICT format is requested either by specifying DSNTIAULSTRICT(NO) as a subparameter of the ULOPTNS parameter or by using the FORMAT DSNTIAUL STRICT option in the UNLOAD command. When you specify CHAR, the output for expressions that have string constants might depend on whether the unload is performed in native or in SQL mode.

VARCHAR

The string constants are handled as VARCHAR data. When you specify VARCHAR, the output for expressions that have string constants does not depend on whether the unload is processed in native or in SQL mode. Consider specifying VARCHAR for unloads that use the DB2 YES option.

Important: If some of your processes use strings that are handled as CHAR data when DB2 HPU processes the unload in native mode, use the default value. Otherwise, specify VARCHAR to ensure that the output of the unload for expressions that have string constants do not depend on the processing mode of the unload.

If you want to specify VARCHAR so that string constants are processed as VARCHAR data in native mode or SQL mode but you have some jobs that expect string constants to be processed as CHAR data, consider changing the

SELECT statements for these unloads by replacing any string constant *string_constant* with CHAR(*string_constant*).

The default value is CHAR.

In previous releases of DB2 HPU, the name of this parameter was VUU064/STRNGCST.

Related concepts:

“DELIMITED format” on page 63

When you create output in the DELIMITED format, you can specify a separator character and a delimiter character.

“VARIABLE format” on page 63

When you create output in the VARIABLE format, the output is compatible with the DB2 LOAD utility input data set.

“EXTERNAL format” on page 65

When you create output in the EXTERNAL format, output fields are in the EXTERNAL format that corresponds to their default type, output records are fixed, and a field separator is not used.

Related reference:

“FORMAT block syntax and description” on page 190

Use the FORMAT block to specify the format of the data that is unloaded. The FORMAT block is a part of the SELECT block.

“DATE format types” on page 459

Use the DATE format type to specify the output data format.

“TIME format types” on page 461

Use the TIME format type to specify the output data format.

“TIMESTAMP format types” on page 461

Use the TIMESTAMP format type to specify the output data format.

DB2 HPU DB2 Administration Tool and DB2 Launchpad parameters

The DB2 Administration Tool and DB2 Launchpad parameters section on the Product Parameters panel (CCQPPRD) in Tools Customizer contains the parameters that are required to integrate DB2 HPU with DB2 Admin.

The following list describes the DB2 HPU DB2 Administration Tool and DB2 Launchpad parameters. The parameters are listed in the following format:

description (Vxxxxnnn/parameter-name)

High-level qualifier for the DB2 Admin data sets

This parameter is optional. It specifies the high-level qualifier of the DB2 Administration Tool libraries. This information is used by the sample programs INZADBI and INZDB2IX (in the SINZCLIST library) to update the DB2 Administration Tool and the Data Management Tools Launchpad tables.

In previous releases of DB2 HPU, the name of this parameter was VUU025.

Library which contains DB2 Admin commands tables

This parameter is optional. It specifies the name of the library that contains the DB2 Administration Tool command tables. This information is used by the sample programs INZADBI and INZDB2IX (in the SINZCLIST library) to update the DB2 Administration Tool and the DB2 Tools Launchpad tables.

In previous releases of DB2 HPU, the name of this parameter was VUU026.

Library which contains the ADBMTI EXEC

This parameter is optional. It specifies the name of the library that contains the ADBMTI EXEC. This information is used by the sample programs INZADBI and INZDB2IX (in the SINZSAMP library) to update the DB2 Administration Tool and the DB2 Tools Launchpad tables.

In previous releases of DB2 HPU, the name of this parameter was VUU027.

DB2 HPU conversion parameters

The DB2 HPU conversion parameters section on the Product Parameters panel (CCQPPRD) in Tools Customizer contains the conversion parameters that are used by DB2 HPU.

The following list describes the DB2 HPU conversion parameters. The parameters are listed in the following format:

description (Vxxxxxxx/parameter-name)

Unicode Conversion Services load library (VZM006/SCUNMOD)

This parameter is optional. It specifies the name of the IBM Conversion Service Load Library. If you want to perform conversions that imply non-SBCS CCSIDs or pairs of SBCS CCSIDs that are not supported by the SYSSTRINGS catalog table, you must first install IBM OS/390® Support for Unicode. For more information about this program, see *Program Directory for z/OS Support for Unicode* and *z/OS Support for Unicode Using Conversion Services*.

In previous releases of DB2 HPU, the name of this parameter was VZM006/SCUNMOD.

Unicode Conversion Services technique search order (VZM007/SCUNTSO)

This parameter is optional. It specifies a list of technique search orders to be searched for the z/OS Unicode Services. Separate each value with a comma. When a conversion between two CCSIDs is required, HPU looks for a conversion that specifies one of the listed technique search orders until it finds one valid conversion.

Example: SCUNTSO = ER,,RE means that DB2 HPU will for the ER technique search order, followed by the *blank* technique search order, and then the RE technique search order.

The default value is blank.

If DB2 Version 8 or later is installed, usually the technique search order is equal to ER.

See *z/OS Support for Unicode Using Conversion Services* for more information.

In previous releases of DB2 HPU, the name of this parameter was VZM007/SCUNTSO.

Unicode Conversion Services substitution character mode (VZM008/SCUNSUB)

This parameter is optional. It specifies how the UNLOAD utility manages conversions through z/OS Unicode Services.

YES Indicates that if a character in a string cannot be converted between two CCSIDs, it is replaced by the substitution character of the converter.

NO Indicates that if a character in a string cannot be converted, the conversion of all the strings fails.

The default value is YES.

In previous releases of DB2 HPU, the name of this parameter was VZM008/SCUNSUB.

DB2 HPU file management parameters

The DB2 HPU file management parameters section on the Product Parameters panel (CCQPPRD) in Tools Customizer contains the file management parameters that are used by DB2 HPU.

The following list describes the DB2 HPU file management parameters. The parameters are listed in the following format:

description (Vxxxxnnn/parameter-name)

Volume name of migrated object (VZM005/INFVSMIG)

This parameter is optional. It specifies the volume name that is located in the ICF catalog for migrated files. The default value is MIGRAT, which corresponds to the value that is used by the DFSMSShsm component of the IBM Data Facility Storage Management Subsystem (DFSMS).

In previous releases of DB2 HPU, the name of this parameter was VZM005/INFVSMIG.

Assignable devices number per tape unit/storage class (VZM010/TAPEUNIT)

This parameter is optional. Use this parameter to specify multiple unit names, storage class names, or both, and the number of associated devices that can be used by DB2 HPU when you request that output files be allocated on tape. DB2 HPU ensures that the maximum number of tape devices is not exceeded by automatically reducing the parallelism degree, if necessary. When the number of tape devices that are associated with a unit name or a storage class name is not specified, the value is set to 1.

This parameter does not have a default value.

The corresponding SYSIN keyword is TAPEUNIT in the GLOBAL OPTIONS block.

In previous releases of DB2 HPU, the name of this parameter was VZM010/TAPEUNIT.

Maximum number of disk units to allocate a work file (VUX010/LIMUNIT)

This parameter is optional. It specifies the maximum number of disk units to be used when DB2 HPU allocates a temporary work file. It is also taken into account to determine the number of work data sets specified as the DYNALLOC first operand of the SORT program when a DB2 HPU needs to perform a sort task.

The default value is 9.

The corresponding SYSIN keyword is LIMUNIT in the Technical Parameters options block, which can be coded in the GLOBAL OPTIONS block.

In previous releases of DB2 HPU, the name of this parameter was VUX010/LIMUNIT.

Note: When allocations are performed by DB2 HPU, the LIMUNIT value is used as the volume count (VOLCNT) allocation parameter unless it exceeds the maximum value accepted by the system for the device type involved. For

example, if LIMUNIT is set to 255 and the works data sets are to be allocated on DASD devices for which the limit is 59, DB2 HPU allocates the work data sets with VOLCNT=59.

Maximum size for primary allocation of a work data set (VUX019/WRKMXPR)

This parameter is optional. It specifies the maximum size, in kilobytes, for the primary allocation of a work data set on DASD. When very large work data sets are used, the primary allocation might be distributed among several volumes according to the limit that was specified in the VUX010/LIMUNIT parameter.

Requirement: Regardless of the limit that was specified on the VUX010/LIMUNIT parameter, the value that you provide for the VUX019/WRKMXPR parameter must be lower than the capacity of the units that is used for these work data sets (VUM013). You must also consider that these units might be distributed among several volumes.

Valid values are 1 - 16777215.

The default value is 500000.

The corresponding SYSIN keyword is WRKMXPR in the Technical Parameters options block, which can be coded in the GLOBAL OPTIONS block.

In previous releases of DB2 HPU, the name of this parameter was VUX019/WRKMXPR.

Note: When DB2 HPU invokes the sort program, the number of work datasets to be used by the sort program is determined according to the following rules:

- The size of a work data set is assumed to be VUX019/WRKMXPR.
- The number of work files per allowed volume (i.e. VUX010/LIMUNIT) must not exceed 2.

These rules are implemented via the following formula: Number of work data sets = min(upper_round(2 * FILSZ * RECORD_LENGTH / WRKMXPR / 1024), 2 * LIMUNIT) where FILSZ stands for the estimated number of records to be sorted and RECORD_LENGTH for the mean value of the record length.

Although the above formula is always used to determine the allowed number of work data sets, the actual number of data sets might be lower if this calculated number exceeds the number of files the sort program can handle. For example, DFSORT V1R12.0 cannot use more than 255. Therefore, although LIMUNIT is set to a value greater than or equal to 128, which would theoretically allow the sort program, to use more than 256 work data sets, no more than 255 files will be actually used.

DSCB model for allocation of GDS (VUX033/TMPLDSCB)

This parameter is optional. It specifies the model data set control block (DSCB) to use when allocating generation data sets (GDS) and when a TEMPLATE is used to allocate new generations of a generation data group (GDG).

In previous releases of DB2 HPU, the name of this parameter was VUX033/TMPLDSCB.

Volumes for allocation of temporary data sets (VUM018/WRKVOL)

This parameter is optional. It specifies the name of the volume where temporary data sets will reside.

In previous releases of DB2 HPU, the name of this parameter was VUM018/WRKVOL.

| **Tape unit where the work data sets must be allocated (VUA007/WRKTUNIT)**

| This parameter is optional. It specifies the name of the tape unit that is used to
| allocate temporary files.

| If you use temporary files on tape, specify a tape unit or DASD device on this
| parameter. If you specify a DASD device, ensure that the pool of volumes that
| are associated with that unit has enough free space to store large data sets. If
| you do not specify a value, the utility allocates temporary files on the unit that
| is specified on the WRKUNIT parameter.

| The corresponding SYSIN keyword is WRKTUNIT of the Technical Parameters
| options block, which can be coded in the GLOBAL OPTIONS block.

| In previous releases of DB2 HPU, the name of this parameter was
| VUA007/WRKTUNIT.

| **Maximum size for work data set on DASD (VUX016/WRKUNTSW)**

| This parameter is optional. It specifies a threshold size (in kilobytes) for work
| data sets. All work data sets that exceed this threshold size will be allocated on
| the unit that is specified on the VUA007/WRKTUNIT parameter.

| The corresponding SYSIN keyword is WRKUNTSW in the Technical
| Parameters options block, which can be coded in the GLOBAL OPTIONS
| block.

| In previous releases of DB2 HPU, the name of this parameter was
| VUX016/WRKUNTSW.

| **Maximum number of unit for tape temporary data set (VUX017/MAXTUNIT)**

| This parameter is optional. It specifies the maximum number of tape units that
| are provided for work data sets that are used by a DB2 HPU job.

| Valid values are 1 - 255.

| The default value is 2.

| The corresponding SYSIN keyword is MAXTUNIT in the Technical Parameters
| options block, which can be coded in the GLOBAL OPTIONS block.

| In previous releases of DB2 HPU, the name of this parameter was
| VUX017/MAXTUNIT.

| **Number of volumes for temporary data set on tape (VUX032/WRKTVcnt)**

| This parameter is optional. It specifies the number of volumes to use for
| temporary data sets that are allocated on tape. Specifying 0 indicates that no
| VOLCOUNT parameter will be used for allocating tape files. In this case, up to
| five volumes are used.

| Valid values are 0 - 255.

| The default value is 0.

| The corresponding SYSIN keyword is WRKTVcnt in the Technical Parameters
| options block, which can be coded in the GLOBAL OPTIONS block.

| In previous releases of DB2 HPU, the name of this parameter was
| VUX032/WRKTVcnt.

| **BUFNO for sequential QSAM (VUM022/QSBUFNO)**

| This parameter is optional. It specifies the number of data buffers for
| sequential QSAM (the BUFNO parameter of the DCB for QSAM).

| Valid values are 1 - 255.

| The default value is 60.

In previous releases of DB2 HPU, the name of this parameter was VUM022/QSBUFNO.

BUFND for sequential VSAM (VUM023/VSBUFND)

This parameter is optional. It specifies the number of data buffers for sequential VSAM (the BUFND parameter of the ACB for VSAM).

Note: Specifying a large value might increase the amount of memory that DB2 HPU requires.

Valid values are 0 - 65535.

The default value is 360.

In previous releases of DB2 HPU, the name of this parameter was VUM023/VSBUFND.

DB2 HPU reporting parameters

The DB2 HPU reporting parameters section on the Product Parameters panel (CCQPPRD) in Tools Customizer contains the reporting parameters that are used by DB2 HPU.

The following list describes the reporting parameters. The parameters are listed in the following format:

description (Vxxxxxxx/parameter-name)

List of system codes for which no dump is produced (VZM009)

This parameter is optional. It defines a list of system codes for which no dump is produced if an abend occurs during execution of DB2 HPU. You can specify up to eight lines. Each line can contain up to eight values. Separate each value by a comma.

After changing the content of this variable, you must regenerate the installation job related to the INZZSCOD template to activate the change.

The list can contain specific system codes that are coded on three hexadecimal digits or generic system codes that begin with X and include all the system codes for the number that follows it. For example, 0c1 is a specific system code, and X78 is a generic system code that includes all system codes that end with 78 (B78, D78, E78, and so on).

The default value is X22, X06, X37, 913, X78, X0A.

In previous releases of DB2 HPU, the name of this parameter was VZM009.

Generate a tape usage report in SYSPRINT (VZM011/TAPERPT)

This parameter is optional. Use it to specify whether to generate a report about the tape usage. The report displays the data set names (DSN) and their positions on the tape (FILESEQ) for each volume that is used.

Specify one of the following values:

YES The tape usage report is generated.

NO The tape usage report is not generated.

The default value is NO.

In previous releases of DB2 HPU, the name of this parameter was VZM011/TAPERPT.

| **Maximum number of messages for row structure errors (VUX018/LDSERRLM)**

| This parameter is optional. It specifies the maximum number of messages that
| are issued if DB2 HPU encounters a row structure error while reading the rows
| of a table space. Use this parameter to limit the number of messages that are
| written into the spool.

| Valid values are 0 - 2147483647.

| **Important:** Specifying a large number of records might increase the amount of
| storage that DB2 HPU requires.

| This parameter does not have a default value.

| In previous releases of DB2 HPU, the name of this parameter was
| VUX018/LDSERRLM.

| **Return code for unloaded rows (VUU024/UNLZLRC)**

| This parameter is optional. It specifies the return code that applies when no
| rows are unloaded by at least one of the SELECT statements of the UNLOAD
| command.

| Valid values are 0 - 4 095.

| The default value 4.

| Alteration applies only when original return code is zero unless keyword
| MAXRC is specified.

| In previous releases of DB2 HPU, the name of this parameter was
| VUU024/UNLZLRC.

| **Return code if an object is missing (VUU024/UNLZLRC)**

| This parameter is optional. It specifies the return code that applies when an
| object included in a LIS TDEF does not exist, or when a LISTDEF expands to
| contain no objects. When this parameter is not specified, no specific return
| code applies for this situation.

| Valid values are 1 - 4095.

| There is no default value.

| Alteration applies only when original return code is zero unless keyword
| MAXRC is specified.

| In previous releases of DB2 HPU, the name of this parameter was
| VUU024/UNLZLRC.

| **Return code if zero line is unloaded (VUU024/UNLZLRC)**

| This parameter is optional. It specifies the return code when a LISTDEF clause
| returns no objects. When this parameter is not specified, no specific return
| code applies for this situation.

| Valid values are 1 - 4095.

| There is no default value.

| Alteration applies only when original return code is zero unless keyword
| MAXRC is specified.

| In previous releases of DB2 HPU, the name of this parameter was
| VUU024/UNLZLRC.

| **MAXRC alteration return code (VUU024/UNLZLRC)**

| This parameter is optional. When MAXRC is not specified, the original return
| code can be altered only when it is 0. When MAXRC is specified, it can also be
| altered when its value is 1 - 8.

| **Frequency to display information messages (VUU050/ULFRQMSG)**

| This parameter is optional. It specifies that DB2 HPU issues informational
| messages that display the current number of unloaded rows for each output
| file.

| *n* At the end of processing, DB2 HPU issues message INZX089, which
| indicates the total number of rows for each output file. A non-null integer
| value means that this message is issued for each *n* rows.

| The default value is 0.

| In previous releases of DB2 HPU, the name of this parameter was
| VUU050/ULFRQMSG.

| **Additional parameter for information messages (VUU050/ULFRQMSG)**

| This parameter is optional. It specifies interval time since the previous message
| was issued or since the first record was unloaded.

| Specify one of the following values:

| **DELTA**

| Messages INZX089 and INZX090 are issued to indicate the elapsed time
| since the previously issued message.

| **TOTAL**

| Messages INZX089 and INZX090 are issued to indicate the total elapsed
| time since the beginning of the unload process.

| TOTAL is the default when a non-null value is specified and neither
| DELTA nor TOTAL is specified.

| TOTAL and DELTA are mutually exclusive.

| The default value is TOTAL.

| In previous releases of DB2 HPU, the name of this parameter was
| VUU050/ULFRQMSG.

| **The location where DB2 HPU writes information messages (VUU051/PROCMSG)**

| This parameter is optional. It specifies where to write informational messages
| that correspond to the unload process of the table space, image copy, or
| partitions and indexes.

| Specify one of the following values:

| **WTO**

| Messages are issued as write to operator messages in the system log.

| **ddname**

| Messages are issued in the corresponding ddname. If the corresponding
| ddname is not already allocated in the JCL, it is dynamically allocated as
| SYSOUT=*. This ddname can be equal to SYSPRINT, in which case these
| messages might be mixed with other messages that DB2 HPU issues.

| All messages that are issued in the output file are prefixed with the system
| time in *hh:mm:ss* format.

| The default value is WTO.

The corresponding SYSIN parameter is OPTIONS PROCMSG.

In previous releases of DB2 HPU, the name of this parameter was VUU051/PROCMSG.

Location of messages issued for each SELECT statement (VUU052/SEMMSG)

This parameter is optional. It specifies where to write informational messages INZX089 and INZX090 that are issued for each SELECT statement.

Specify one of the following values:

WTO

Messages that correspond to the SELECT statements are issued as write to operator messages in the system log.

NUMBERED

Messages that correspond to the SELECT statements are issued in a ddname for each SELECT statement. If this ddname is not already allocated in the JCL, it is dynamically allocated as SYSOUT=*. The format of the generated ddname is UxxSyyyy, where xx is the two-digit UNLOAD number and yyyy is the SELECT number for this unload.

ddname

Messages that correspond to the SELECT statements are issued in the corresponding ddname. If this ddname is not already allocated in the JCL, it will be dynamically allocated as SYSOUT=*. This ddname can be equal to SYSPRINT, in which case these messages might be mixed with other messages that DB2 HPU issues.

The default value is WTO.

The corresponding SYSIN parameter is OPTIONS SELMSG.

In previous releases of DB2 HPU, the name of this parameter was VUU052/SELMSG.

Generate a list of all OBIDs in the FIC (VUU053/OBIDRPT)

This parameter is optional. It specifies whether to generate a list of all object IDs (OBIDs) that are found in the full image copy (FIC). It applies only when you are unloading from an image copy.

YES

Generates a list of all OBIDs that are found in the FIC.

NO Does not generate a list of all OBIDs that are found in the FIC.

The default value is NO.

The corresponding SYSIN parameter is OBID_REPORT in the COPYDDN option.

In previous releases of DB2 HPU, the name of this parameter was VUU053/OBIDRPT.

Return code when the utility has switched to SQL mode (VUU069/SQLSWIRC)

This parameter is optional. It specifies the return code that applies when the processing of a select statement has automatically switched to SQL mode when the statement is not supported in native mode. It is not applicable when the SQL mode is explicitly requested (e.g. DB2 HPU FORCE is specified).

Valid values are 0 - 4 095.

The default value is 4.

DB2 HPU tuning parameters

The DB2 HPU tuning parameters section on the Product Parameters panel (CCQPPRD) in Tools Customizer contains the tuning parameters that are used by DB2 HPU.

The following list describes the tuning parameters. The parameters are listed in the following format:

description (Vxxxxnnn/parameter-name)

Default size of the input file or object for DB2 HPU (VUX003/SIZE)

This parameter is optional. It specifies the default number of records when allocating the resources (work data sets and FILSZ parameter) that are required by the SORT processing. The default value is used only when DB2 HPU is unable to estimate the number of rows to be unloaded, which happens only in very specific cases.

This parameter is ignored when the rows are selected by DB2. Therefore, set VUX003/SIZE to a typical or maximum number of rows for the table spaces that are usually processed by DB2 HPU. VUX003/SIZE is expressed as a number of records.

Valid values are 1000 - 2147483647

Important: Specifying a large number of records might increase the amount of storage that DB2 HPU requires.

The default value is 1000000 records.

In previous releases of DB2 HPU, the name of this parameter was VUX003/SIZE.

Minimum memory size for each sort process (VUX004/LOWMEM)

This parameter is optional. It specifies the memory size, in bytes, below the 16 MB line that is used by the sort process. DB2 HPU considers LOWMEM only during parallel processing. When DB2 HPU prepares to start one or more SORT tasks (with the exception of the first SORT), it checks whether the amount of remaining memory under the line is greater or equal to LOWMEM.

Take the following into account in order to determine the MAXSORT and LOWDEM parameters:

1. Determine the maximum storage that is allowed for a SORT below the 16 MB limit; this value is referred to as SORTMAXMEMBELOW hereafter. For example, SORTMAXMEMBELOW is the value of VSCORE parameter of the DFSORT installation and the value of VSCORE parameter of the SYNC SORT installation. Refer to the documentation of your SORT program to determine the SORTMAXMEMBELOW value if you use another one.
2. Determine the minimum amount of memory required below the 16 MB line by your SORT program: This value is referred to as SORTMINMEMBELOW hereafter. To do so, either refer to your SORT program documentation or run a sample of SORT program sorting no rows or a very little number of rows and check the amount of memory this program has used below the 16MB line (see the SYS value given by the IEF371I message). SORTMINMEMBELOW is about 2070 000 bytes for DFSORT VR1R10.
3. Determine the remaining memory below the 16 MB limit when the INZUTILB module is loaded in memory: this value is referred to as HPUMEMBELOW hereafter. To do so, run a sample DB2 HPU job requesting a basic unload (the IVP job is suitable) with the QUIESCE YES

option and no SORT request. Check the amount of memory this program has used below the 16 MB line (see the SUS value given by the IEF374I message).

Set the LOWMEM and MAXSORT according to the following rules:

- $\text{SORTMINMEMBELOW} < \text{LOWMEM} < \text{SORTMAXMEMBELOW}$
- $1 < \text{MAXSORT} < (16 \text{ MB} - \text{HPUMEMBELOW}) / \text{LOWMEM}$

To prevent failures, do not set LOWMEM lower than SORTMINMEMBELOW.

To maximize the amount of memory that is used, set LOWMEM to a value less than or equal to the value of SORTMAXMEMBELOW. Setting LOWMEM to a higher value would lead of a waste of memory equal to $\text{LOWMEM} - \text{SORTMAXMEMBELOW}$. Setting LOWMEM equal to LOWMAXMEMBELOW ensures that any SORT task launched by DB2 HPU can use an amount of memory up to that value of SORTMAXMEMBELOW. However, as SORTMAXMEMBELOW might not always be needed,, consider setting LOWMEM to a lower value so that DB2 HPU will attempt to launch more SORT tasks. Do not decrease LOWMEM to a too low value as it can lead to failures because some SORT tasks might run out of memory. The default setting (in parmlib) might not be relevant for some complex unload jobs where alot of tasks are involved and parallelism is enabled. In such a case, it is recommended to adjust the value of LOWMEM and MAXSORT via the Technical Parameters option block in the GLOBAL block in SYSIN.

To improve performance, increase MAXSORT and decrease LOWMEM.

To minimize the consumption of memory below the 16 MB line, decrease MAXSORT and increase LOWMEM.

Valid values are 1 - 2147483647.

The default value is 270000.

The corresponding SYSIN keyword is LOWMEM in the Technical Parameters options block, which can be coded in the GLOBAL OPTIONS block.

In previous releases of DB2 HPU, the name of this parameter was VUX004/LOWMEM.

Wait unit in seconds/100 (VUX007/WAITUNIT)

This parameter is required. It specifies the wait time (in hundredths of seconds) between two unsuccessful tests of the STOP command.

Valid values are 1 - 2147483647.

The default value is 100 (1 second).

In previous releases of DB2 HPU, the name of this parameter was VUX007/WAITUNIT.

Number of wait periods before issuing a WTOR (VUX008/WAITQTY)

This parameter is required. It specifies the number of times an unsuccessful STOP command will be tested before a WTOR is sent to the console. If the operator answers CANCEL (C), DB2 HPU will stop with return code 8. If the operator answers WAIT (W), the wait process starts again.

Valid values are 1 - 2147483647.

The default value is 20.

In previous releases of DB2 HPU, the name of this parameter was VUX008/WAITQTY.

|
| **Wait time for the WTOR reply (VUX009/WAITQTYM)**

| This parameter is required. It specifies the maximum wait time (in seconds)
| before an answer is sent to the WTOR message. Utility execution will stop
| beyond this limit (return code 8).

| Valid values are 1 - 2147483647.

| The default value is 60.

| In previous releases of DB2 HPU, the name of this parameter was
| VUX009/WAITQTYM.

| **Size of work areas (VUX022/VBUFSIZE)**

| This parameter is optional. It sets the size, in bytes, of the buffers that are used
| for communication between DB2 HPU tasks such as reading LDS, formatting
| data, and sorting data. The value that you set depends on the speed of the
| system. A higher value has little impact on performance, but a lower value
| forces DB2 HPU to change tasks more frequently, which increases WAIT TIME
| and CPU TIME.

| Valid values are 1 - 214748 3647.

| The recommended value is 1000000. Do not change VBUFSIZE unless
| performance problems occur.

| The default value is 1000000.

| The corresponding SYSIN keyword is VBUFSIZE in the Technical Parameters
| options block, which can be coded in the GLOBAL OPTIONS block.

| In previous releases of DB2 HPU, the name of this parameter was
| VUX022/VBUFSIZE.

| **Maximum degree of parallelism for LDS reading (VUX025/PARALLEL)**

| This parameter is optional. It specifies the parallelism degree for an unload job
| when you are unloading a partitioned table space. This parameter indicates the
| maximum number of partitions that are processed in parallel. However, if the
| unload job includes a sort process, the VUX025 parameter is limited to the use
| of the VUX005/MAXSORT parameter. When a sort process is included, if the
| value of the VUX005/MAXSORT parameter is lower than the value that is
| specified in the VUX025/PARALLEL parameter, the VUX005/MAXSORT
| parameter is used. You can specify a value of 1 or greater.

| Valid values are 1 - 65535.

| The default value is 20.

| In previous releases of DB2 HPU, the name of this parameter was
| VUX025/PARALLEL.

| **Maximum degree of parallelism between unload tasks (VUX030/UTLPARAL)**

| This parameter is optional. It specifies the parallelism degree when you are
| starting several unload tasks for the same UNLOAD command. An unload
| task can be when you unload a table space (each table space that is accessed
| counts for one task, whether it is partitioned or not), or a SELECT statement
| that is processed by DB2. If more than one SELECT statement is processed by
| using DB2, DB2 HPU will count one task for all of these SELECT statements.
| You can specify a value of 1 or greater.

| Valid values are 1 - 65535.

| The default value is 5.

In previous releases of DB2 HPU, the name of this parameter was VUX030/UTLPARAL.

Number of rows retrieved by an SQL FETCH function (VUU035/ULROWSET)

This parameter is optional. It specifies the number of rows that will be retrieved by a single SQL FETCH. Specifying the number of rows that will be retrieved improves the performance for SELECT statements that are processed by DB2 (using DB2 FORCE or DB2 YES with unsupported SELECT statements). When you specify 1, the multi-fetch function is not used. A standard single-row FETCH is used instead. This function is available only in DB2 Version 8 and later releases.

Valid values are 1 - 32767.

The default value is 100.

In previous releases of DB2 HPU, the name of this parameter was VUU035/ULROWSET.

Enforce partition parallelism for unloading a table into a single file (VUU036/GBLPARAL)

This parameter is optional. It specifies whether partition parallelism will be enforced when a table space is unloaded into a single output file.

Specify one of the following values:

YES

Parallelism is enforced when table spaces are unloaded. The parallelism degree is set in variables VUX025/PARALLEL and VUX005/MAXSORT when a sort is requested.

NO Parallelism is determined by the number of output files that are coded in the JCL and the explicit selection of partitions in the SYSIN. You can override this PARMLIB variable by using the PARALLELISM SYSIN keyword (DB2 HPU syntax only).

The default value is NO.

In previous releases of DB2 HPU, the name of this parameter was VUU036/GBLPARAL.

Support parallelism for unloading several partitions (VUU044/SUBTKSOF)

This parameter is optional. It specifies whether parallelism is supported when data is unloaded from several partitions into a single output data set.

Specify one of the following values:

NO Does not support parallelism when data is unloaded from several partitions into a single output data set. NO is the recommended value.

YES

Supports parallelism when data is unloaded from several partitions into a single output data set.

The default value is NO.

Attention: Do not change this value unless IBM asks you to change it.

In previous releases of DB2 HPU, the name of this parameter was VUU044/SUBTKSOF.

Maximum number of partitions processed in one unload (VUU060/ULMAXPAR)

This parameter is optional. It specifies the maximum number of partitions that

DB2 HPU can process in a single unload operation. ULMAXPAR affects logical unload operations (SELECT) that are processed natively and physical unload operations only. You can use this parameter to reduce the memory consumption of DB2 HPU.

If the total number of partitions that have to be processed for a table space is larger than the value that is specified in ULMAXPAR, DB2 HPU automatically splits the single unload operation into several unload operations to respect the value of ULMAXPAR. If the value of ULMAXPAR is smaller than the value of PARALLELISM at the partition level, the effective partition parallelism is limited by ULMAXPAR.

Valid values are 0 - 4096.

The default value is 0, which means that splitting is not done. When you specify a non-zero value, ORDER BY and ORDER CLUSTER clauses can be used only when each partition is unloaded into a separate file. When you specify a non-zero value and a split is done, the OUTMAXROWS or UNLMAXROWS setting applies to each partition.

The corresponding SYSIN keyword is MAXPART, which can be specified in the GLOBAL OPTIONS block or the UNLOAD block.

In previous releases of DB2 HPU, the name of this parameter was VUU060/ULMAXPAR.

Size of the buffer used to retrieve LOB data (VUU041/ULOCSIZE)

This parameter is optional. It specifies the size of the buffer, in bytes, to be used to retrieve LOB data by using a LOB locator through DB2. This parameter is used in DB2 Versions 7 and 8 when retrieving LOB data by using LOBFILE REFERENCE (CLOBF, BLOBF, or DBCLOBF) in DB2 FORCE or DB2 YES with an unsupported SELECT statement. In DB2 Version 9, DB2 HPU uses LOB FILE REFERENCE variables and does not require intermediate buffers.

Valid values are 1 - 16000000.

The default value is 1000000.

In previous releases of DB2 HPU, the name of this parameter was VUU041/ULOCSIZE.

Wait/retry function when resources are unavailable (VUU047/RETRYMOD)

This parameter is optional. If a resource is not available, it specifies whether DB2 HPU uses wait/retry logic. This parameter controls the following availability criteria:

- The status of the DB2 object (table space, partition, or index) that prevents processing
- If LOCK YES is specified in the SYSIN and a LOCK cannot be taken on the corresponding object
- If the dynamic allocation of the LDS files of the table space or index to be unloaded fails

Specify one of the following values:

NONE

DB2 HPU does not use wait/retry logic.

ALL

DB2 HPU uses wait/retry logic for the three kinds of situations in the previous list.

STATUS

DB2 HPU uses wait/retry logic when the status of the DB2 object, table space, partition, or index prevents processing. The following statuses prevent processing:

- CHKP
- GRECP
- UTUT
- PSRBD
- WEPR
- RBDP
- RBDP*
- REORP
- DBETE
- RESTP
- PSRCP
- LPL
- RELDP
- REFP
- RECP
- RECP*

LOCK

DB2 HPU uses wait/retry logic when a LOCK is requested by the LOCK YES option and the LOCK cannot be taken.

LDS

DB2 HPU uses wait/retry logic when LDS allocation fails.

You can specify multiple values except when you specify ALL or NONE. If you specify multiple values, separate them by using commas without spaces. Specify the wait time (in hundredths of a second) between two unsuccessful tests of the resource availability with PARMLIB parameter VUU048/RETRYW, and specify the number of retries with PARMLIB parameter VUU049/RETRYNB.

The default value is NONE.

In previous releases of DB2 HPU, the name of this parameter was VUU047/RETRYMOD.

Wait time between tries to access unavailable resources (VUU048/RETRYW)

This parameter is optional. It specifies the wait time (in hundredths of seconds) between two unsuccessful tests of the resource availability that is defined by the VUU047/RETRYMOD parameter.

Valid values are 1 - 32767.

The default value is 100 (1 second).

In previous releases of DB2 HPU, the name of this parameter was VUU048/RETRYW.

Number of retries when resources are unavailable (VUU049/RETRYNB)

This parameter is optional. It specifies the number of times to test the availability of a resource before stopping with return code 8.

Valid values are 1 - 32767.

The default value is 20.

In previous releases of DB2 HPU, the name of this parameter was VUU049/RETRYNB.

Concurrent access resolution for prepared statement (VUU061/ACCPREP)

This parameter is optional. Use it to specify the CONCURRENT ACCESS RESOLUTION option for the PREPARE statement that DB2 HPU will use to process SELECT statements by using DB2.

Specify one of the following values:

WAIT

The WAIT FOR OUTCOME clause is specified to wait for the commit or rollback of data that is being updated or deleted.

USE

The USE CURRENTLY COMMITTED clause is specified to use the currently committed version of the data that is being updated or deleted. The USE option is supported only in DB2 10 and later releases.

The VUU061 parameter does not have a default value.

In previous releases of DB2 HPU, the name of this parameter was VUU061/ACCPREP.

DB2 HPU sort parameters

The DB2 HPU sort parameters section on the Product Parameters panel (CCQPPRD) in Tools Customizer contains the sort parameters that are used by DB2 HPU.

The following list describes the common sort parameters. The parameters are listed in the following format:

description (Vxxxxxx/parameter-name)

Maximum number of active sorts processes (VUX005/MAXSORT)

This parameter is optional. It specifies the maximum number of active sorts that can run in the same step when parallel processing is involved. IBM DFSORT for z/OS supports MAXSORT values 1 - 20, and all of its modules are reusable.

Most of the non-IBM sort products contain modules that require MAXSORT=1 because not all of their modules are reusable.

To improve performance, increase MAXSORT and decrease LOWMEM.

To minimize the consumption of memory below the line, decrease MAXSORT and increase LOWMEM.

Valid values are 1 - 32767.

The default value is 20.

The corresponding SYSIN keyword is MAXSORT in the Technical Parameters options block, which can be coded in the GLOBAL OPTIONS block.

In previous releases of DB2 HPU, the name of this parameter was VUX005/MAXSORT.

| **Allocation distribution for the sort input file (VUX006/WRKSPACE)**

| This parameter is optional. It specifies the percentage of space that is allocated
| for the sort input file. WRKSPACE can be one of the following values:

| Specify one of the following values:

| **PARTIAL**

| Corresponds to a primary and secondary allocation that is equal to 50
| percent of the estimated size of the file to be sorted. Two volumes are
| allowed for this allocation.

| **FULL**

| Corresponds to a primary allocation that is equal to 100 percent of the
| estimated size of the file to be sorted, and to a secondary allocation of 33
| percent.

| The default value is PARTIAL.

| The corresponding SYSIN keyword is WRKSPACE in the Technical Parameters
| options block, which can be coded in the GLOBAL OPTIONS block.

| In previous releases of DB2 HPU, the name of this parameter was
| VUX006/WRKSPACE.

| **Sort program that DB2 HPU uses to sort data (VUX037/SORTUTIL)**

| This parameter is optional. It allows you to specify which sort program to use
| when DB2 HPU needs to perform a sort operation outside of DB2.

| Specify one of the following values:

| **SORT**

| The on-site sort program is used. The load module name is DFSORT.

| **DB2SORT**

| The IBM DB2 Sort for z/OS utility (DB2 Sort) is used. DB2 Sort can
| improve the performance of utility sort processing, especially in
| environments with large volumes of data, large table spaces, large indexes,
| or batch window constraints. These environments require a more
| sophisticated sorting approach than the approach that is used by tools that
| are used for general sorting purposes.

| The load module name is DB2SORT. The relevant entry point load module
| of DB2 Sort must be installed in the system libraries. Specifying DB2SORT
| does not affect the DB2 settings. Therefore, if you want the sorts performed
| by DB2 (for unsupported SELECT statements or when DB2 FORCE is
| specified), you must change the appropriate DB2 settings.

| **Tip:** Specify DB2SORT when you need to perform several sort operations
| in parallel.

| For more information about DB2 Sort, see the *DB2 Sort for z/OS User's
| Guide*.

| The default value is SORT.

| The corresponding SYSIN keyword is SORTUTIL in the GLOBAL OPTIONS
| block.

| In previous releases of DB2 HPU, the name of this parameter was
| VUX037/SORTUTIL.

| **NNumber of records in sort work areas (VUM024/SRTNBVRE)**

| This parameter is optional. It specifies the number of records in the sort work
| areas.

| **Important:** Specifying a large number of records might increase the amount of
| storage that DB2 HPU requires.

| Valid values are 0 - 2147483647.

| The default value is 800.

| In previous releases of DB2 HPU, the name of this parameter was
| VUM024/SRTVNBRE.

| **Minimum size in bytes for sort work areas (VUM025/SRTVSMIN)**

| This parameter is optional. It specifies the minimum size, in bytes, of the sort
| work areas.

| **Important:** Specifying a large value might increase the amount of storage or
| memory that DB2 HPU requires.

| Valid values are 0 - 2147483647.

| The default values are the values that are set for VUX022.

| The corresponding SYSIN keyword is SRTVSMIN in the Technical Parameters
| options block, which can be coded in the GLOBAL OPTIONS block.

| In previous releases of DB2 HPU, the name of this parameter was
| VUM025/SRTVSMIN.

| **Maximum size for sort work areas (VUM026/SRTVSMAX)**

| This parameter is optional. It specifies the maximum size, in bytes, of the sort
| work areas.

| **Note:** Specifying a large value might increase the amount of memory that DB2
| HPU requires.

| Valid values are 0 - 2147483647.

| The default value is 2000000.

| The corresponding SYSIN keyword is SRTVSMAX in the Technical Parameters
| options block, which can be coded in the GLOBAL OPTIONS block.

| In previous releases of DB2 HPU, the name of this parameter was
| VUM026/SRTVSMAX.

| **Customization templates**

| Tools Customizer uses the following templates to generate the customization jobs
| for DB2 HPU.

| The following table lists and describes the templates:

| *Table 35. Template names and descriptions*

Template	Description
INZTVAR	Generates the INZUTIL member containing the non DB2-dependent settings
INZPARM	Defines the dsname of the DB2 HPU PARMLIB to the load modules

Table 35. Template names and descriptions (continued)

Template	Description
INZHPUCL	Generates the members required to run the DB2 HPU ISPF interface
INZLAUNC	Generates the clist to be run to add DB2 HPU to DB2 Tools Launchpad
INZADTOO	Generates a member with instructions for adding DB2 HPU into DB2 Admin
INZBIND	Binds the plans and packages that are used by DB2 HPU and grants access
INZEXEUE	Generates a sample job for running DB2 HPU against a given subsystem
INZZSCOD	Defines the list of abend codes trapped for dump
INZCHECK	Generates members with the DB2-dependent settings and runs DB2 HPU IVP
INZFREE	Frees the DB2 HPU plan for a DB2 subsystem or a data sharing group

INZDB21X member

Use the INZDB21X member to integrate DB2 HPU with DB2 Administration Tool.

The following example shows the INZDB21X member. It is not customized.

```

/*-----*/
/*                                          */
/* Member : INZDB21X                      */
/*                                          */
/*-----*/
/*
 * 5655-AA1
 * (c) Copyright Infotel 1996, 2010 All Rights Reserved.
 */
/*-----*/
/*                                          */
/* This member describes the instructions to apply in order to
/* integrate DB2 HPU into the DB2 Administration Tool.
 */
/*-----*/
/*-----*/
/* INSTRUCTIONS :
/* HPU installation :Please read all the procedure before beginning.
 */
/*-----*/
/*-----*/
/* 1) FIRST PART : make a backup
 */
/*-----*/
/*-----*/
/* a) Please back up the following db2 rexx procedures before
/* beginning any modification :
 */
/*
   - &VUU027(ADB21S)
   - &VUU027(ADB21T)
 */
/*
 */
/* INSTRUCTIONS :
/* HPU installation :Please read all the procedure before beginning.
 */
/*-----*/
/*-----*/
/* 1) FIRST PART : make a backup
 */
/*-----*/
/*-----*/
/* a) Please back up the following db2 rexx procedures before
 */

```



```

/* beginning any modification : */
/*
- &VUU027(ADB21S)
- &VUU027(ADB21T)
*/

ADDRESS ISPEXEC "LIBDEF ISPTABL DATASET" ,
  "ID('&VUU026')",
  "STACK"
ADDRESS ISPEXEC "LIBDEF ISPTLIB DATASET" ,
  "ID('&VUU026')",
  "STACK"
**/
/* If you do so, the two following lines must be added at the end of */
/* the rexx procedures before the exit command */
/*
ADDRESS ISPEXEC "LIBDEF ISPTABL DATASET"
ADDRESS ISPEXEC "LIBDEF ISPTLIB DATASET"
**/
/*-----*/
/*-----*/
/* 2) SECOND PART : update the ADB21S table. */
/* ( Command lines table for the DB2ADMIN tablespace panel ) */
/*-----*/
/*-----*/
/* Look for HPU in the file */
/*
&VUU027(ADB21S)

/* If it is not in, then, after backing up this file, insert the */
/* following lines. */
/* These lines should be added before the TBSORT and TBCLOSE */
/* command, near the end of the file. Be careful not to insert */
/* these lines before the TBADD statement of another command. */
/*
ACMD=''
PAN=''
SQL=''
CMD='HPU'
DESCRIP='Additional command - High Performance Unload'
ISPF='SELECT CMD(INZHPU D TS &&DB2N &&DBNAME &&NAME)' "TBADD" table
**/
/*-----*/
/*-----*/
/* 3) THIRD PART : update the ADB21T table */
/* ( Command lines table for the DB2ADMIN table panel ) */
/*-----*/
/*-----*/
/* Look for HPU in the file */
/*
&VUU027(ADB21T)

/* If it is not in, then, after backing up this file, insert the */
/* following lines. */
/* These lines should be added before the TBSORT and TBCLOSE */
/* command, near the end of the file. Be careful not to insert */
/* these lines before the TBADD statement of another command. */
/*
ACMD=''
PAN=''
SQL=''
CMD='HPU'
DESCRIP='Additional command - High Performance Unload'
ISPF='SELECT CMD(INZHPU D TB &&DB2N &&DBNAME &&TSNAME &&CREATOR &&NAME)'
"TBADD" table
**/
/*
/* If APAR PK15597 is applied, replace the above */
/* "TBADD" table' statement with the following 4 statements: */
/*
"TBADD" tableT
"TBADD" tableG
"TBADD" tableM
"TBADD" tableV
**/
/*-----*/
/*-----*/
/* 4) LAST PART : execute the updated rexx procedures */
/*-----*/
/*-----*/
/* This modifications will be taken into account only if the rexx */

```

```

/* procedures, ADB21S and ADB21T, are executed. */
/*-----*/

```

This example contains the following variables:

&VUU026

The library that contains the DB2 Admin Tool commands tables, such as DBTOOL.SADBTLIB.

&VUU027

The library that contains the ADBDMTI EXEC, such as DBTOOL.SADBEXEC.

Related tasks:

“Optional: Integrating DB2 HPU into DB2 Administration Tool” on page 55
 Optionally, you can integrate DB2 HPU into DB2 Administration Tool (DB2 Admin). Tools Customizer will create the necessary JCL, but you must manually complete some steps after you submit the customization job.

Data types for output (TYPE keyword)

The TYPE keyword of the SELECT statement (OPTION block for FORMAT USER) is used to create several types of data in the output. These types are declared in the keyword TYPE. The use of this keyword implies that data is to be converted from the original column type to the type that is declared in the TYPE keyword.

The output data types that are allowed are described in the following topics:

- “Numeric data”
- “Nonnumeric data” on page 459
- “DATE, TIME, and TIMESTAMP data” on page 459
- “Supported conversions” on page 462

Numeric data

The numeric data type declared in the TYPE keyword is used to specify the output numeric data type.

You can use the following formats to code numeric data:

INTEGER or INT

Whole numbers in a binary word of 31 bits plus the sign.

SMALLINT

Whole numbers in a binary halfword of 15 bits plus the sign.

DECIMAL(*n,m*) or DEC(*n,m*)

Standard decimal value that is contained in (*n*/2+1) bytes. The default value is DECIMAL or DEC and is equivalent to DECIMAL (5,0).

DECIMAL ZONED(*len,scale*)DECIMAL ZONED(*len*)DECIMAL ZONED

The format of a zoned number is *znznzn...z/sn*, where *n* denotes a 4-bit decimal digit (called the numeric bits); *z* is the digit's zone (the left four bits of a byte); *s* is the right-most operand that can be a zone (*z*) or a sign value (hexadecimal A, C, E, or F for a positive number, and hexadecimal B or D for a negative number).

len Specifies the number bytes (decimal digits) in the output field. The length must be 1 - 31. If the source data type is DECIMAL and the

length parameter is omitted, the default length is determined by the column attribute that is defined in the table. Otherwise, the default length is 31 bytes.

scale Specifies the number of digits to the right of the decimal point (a decimal point is not included in the output field in this case). There must be an integer that is greater than or equal to zero and must be less than or equal to the length. The default depends on the column attributes that are defined in the table. If the source data type is DECIMAL, the defined scale value is taken as the default value; otherwise, the default is 0.

An error occurs if the output field size that is specified is less than the data length. If the size of the output field is less than the data length, 'X'F0' is padded on the left.

FLOAT(*n*)

Number (*n*) is simple floating point precision if ($0 < n < 22$) in a fullword. Number is double floating point precision if ($21 < n < 54$).

The default type for FLOAT is double precision.

Nonnumeric data

The nonnumeric data type declared in the TYPE keyword is used to specify the output nonnumeric data type.

You can use the following formats to code nonnumeric data:

CHARACTER(*n*) or CHAR(*n*)

Character string of length *n* ($0 < n < 255$) bytes.

VARCHAR(*n*)

A two-byte length field that is followed by *n* characters. The size equals *n*+2 bytes. The DB2 type LONG VARCHAR is not used in a sequential data set.

GRAPHIC(*n*)

Graphic character string that is coded on $2n$ bytes. One character equals 2 bytes.

VARGRAPHIC(*n*)

Variable-length graphic character string that is coded on $2n+2$ bytes. The DB2 type LONG VARGRAPHIC is not used in a sequential data set.

DATE, TIME, and TIMESTAMP data

The DATE, TIME, and TIMESTAMP data types that are declared in the TYPE keyword are used to specify the output data types.

DATE format types

Use the DATE format type to specify the output data format.

The formats, data types, and default length for DATE are shown in the following table:

Table 36. DATE format types

Format	Output data	Data type	Default length (bytes)
DATE_A	MM-DD-YYYY	Character	10

Table 36. DATE format types (continued)

Format	Output data	Data type	Default length (bytes)
DATE_B	MM-DD-YY	Character	8
DATE_C	YYYY-MM-DD	Character	10
DATE_D	YY-MM-DD	Character	8
DATE_E	DD-MM-YYYY	Character	10
DATE_F	DD-MM-YY	Character	8
DATE_G	YYYY-DDD	Character	8
DATE_H	YY-DDD	Character	6
DATE_I	MMDDYYYY	Character	8
DATE_J	MMDDYY	Character	6
DATE_K	YYYYMMDD	Character	8
DATE_L	YYMMDD	Character	6
DATE_M	DDMMYYYY	Character	8
DATE_N	DDMMYY	Character	6
DATE_O	YYYYDDD	Character	7
DATE_P	YYDDD	Character	5
DATE_Q	YYDDMM	Zoned decimal	6
DATE_R	YYYYDDMM	Zoned decimal	8
DATE_0	YYYYMMDD	Packed decimal	5
DATE_1	MMDDYY	Packed decimal	4
DATE_2	DDMMYY	Packed decimal	4
DATE_3	YYMMDD	Packed decimal	4
DATE_4	YYDDMM	Packed decimal	4
DATE_5	MMDDYYYY	Packed decimal	5
DATE_6	DDMMYYYY	Packed decimal	5
DATE_7	YYYYDDMM	Packed decimal	5
DATE_DB2	Date format of the site	Character	10 or the length of the LOCAL date format

By default, DB2 HPU uses the hyphen character (-) or content of the PARMLIB variable VUU055/DATEDDEL as a delimiter. To change this delimiter, code DATEDELIM 'd', where *d* is the new delimiter, in the OPTIONS block.

Related reference:

“OPTIONS block syntax and description” on page 134

Use the OPTIONS block to specify the default conversions that are with the SELECT statements. This block can be used in the GLOBAL block, the UNLOAD block, and the SELECT block.

“DB2 HPU output data parameters” on page 410

The DB2 HPU output data parameters section on the Product Parameters panel (CCQPPRD) in Tools Customizer contains the parameters for configuring output data.

TIME format types

Use the TIME format type to specify the output data format.

The formats, data types, and default length for TIME are shown in the following table:

Table 37. TIME format types

Format	Output data	Data type	Default length (bytes)
TIME_A	HH.MM.SS	Character	8
TIME_B	HH.MM	Character	5
TIME_C	HH.MM AM	Character	8
TIME_D	HHMMSS	Character	6
TIME_E	HHMM	Character	4
TIME_0	HHMMSS	Packed decimal	4
TIME_DB2	Time format of the site	Character	8 or the length of the LOCAL time format

By default, DB2 HPU uses the '.' delimiter character or content of PARMLIB variable VUU056/TIMEDEL. To change the delimiter, code TIMEDELIM 'd', where *d* is the new delimiter, in the OPTIONS block.

Related tasks:

"Selecting time formats" on page 252

The Type Values - TIME panel is used to select a time format.

Related reference:

"OPTIONS block syntax and description" on page 134

Use the OPTIONS block to specify the default conversions that are with the SELECT statements. This block can be used in the GLOBAL block, the UNLOAD block, and the SELECT block.

"DB2 HPU output data parameters" on page 410

The DB2 HPU output data parameters section on the Product Parameters panel (CCQPPRD) in Tools Customizer contains the parameters for configuring output data.

TIMESTAMP format types

Use the TIMESTAMP format type to specify the output data format.

The formats, data types, and default length for TIMESTAMP are shown in the following table:

Table 38. TIMESTAMP format types

Format	Output data	Data type	Default length (bytes)
TIMESTAMP_A	YYYY-MM-DD-HH.MM.SS	Character	19
TIMESTAMP_B	YYYY-MM-DD-HH.MM.SS.NNN...NNN	Character	19 - 31 ¹
TIMESTAMP_C	YYYYMMDDHHMMSS	Character	14
TIMESTAMP_D	YYMMDDHHMMSS	Character	12
TIMESTAMP_E	YYYYMMDDHHMMSSNNN...NNN	Character	14 - 26 ¹
TIMESTAMP_F	YYMMDDHHMMSSNNN...NNN	Character	12 - 24 ¹

Table 38. *TIMESTAMP* format types (continued)

Format	Output data	Data type	Default length (bytes)
TIMESTAMP_G	YYYY-MM-DD HH:MM:SS.NNN	Character	23
TIMESTAMP_0	YYYYMMDDHHMMSSNNN...NNN	Packed decimal	8 - 14 ¹

Note:

1. The default length depends on the precision of the input data type.

Related tasks:

“Selecting timestamp formats” on page 253

The Type Values - *TIMESTAMP* panel is used to select a timestamp format.

Related reference:

“*OPTIONS* block syntax and description” on page 134

Use the *OPTIONS* block to specify the default conversions that are with the *SELECT* statements. This block can be used in the *GLOBAL* block, the *UNLOAD* block, and the *SELECT* block.

“DB2 HPU output data parameters” on page 410

The DB2 HPU output data parameters section on the Product Parameters panel (CCQPPRD) in Tools Customizer contains the parameters for configuring output data.

Supported conversions

DB2 HPU supports conversions to specific output data types.

The following table shows all the supported conversions for DB2 data types and DB2 HPU data types. Conversions that are not contained in this table are not supported by DB2 HPU.

Table 39. *Supported conversions*

DB2 data type	Output data types
INTEGER	SMALLINT, DECIMAL, FLOAT, CHAR
SMALLINT	INTEGER, DECIMAL, FLOAT, CHAR
DECIMAL(<i>m,n</i>)	SMALLINT, INTEGER, CHAR, DECIMAL(<i>p,q</i>), FLOAT
FLOAT	SMALLINT, INTEGER, CHAR, DECIMAL, FLOAT
CHAR(<i>n</i>)	CHAR(<i>m</i>), VARCHAR(<i>m</i>)
VARCHAR(<i>n</i>)	CHAR(<i>m</i>), VARCHAR(<i>m</i>)
LONG VARCHAR(<i>n</i>)	CHAR(<i>m</i>), VARCHAR(<i>m</i>)
GRAPHIC(<i>n</i>)	GRAPHIC(<i>m</i>), VARGRAPHIC(<i>m</i>), CHAR(<i>m</i>)
VARGRAPHIC(<i>n</i>)	GRAPHIC(<i>m</i>), VARGRAPHIC(<i>m</i>)
LONG VARGRAPHIC(<i>n</i>)	GRAPHIC(<i>m</i>), VARGRAPHIC(<i>m</i>)

Syntax compatibility

DB2 HPU provides limited support for Fast Unload syntax and UNLOAD PLUS syntax.

Topics:

- “Fast Unload syntax compatibility”
- “UNLOAD PLUS syntax compatibility” on page 469
- “DB2 unload compatibility” on page 476

Fast Unload syntax compatibility

DB2 HPU provides limited support for the JCL that is used with Fast Unload for DB2, Version 3.1; however, some features of the Fast Unload product might be ignored or be interpreted differently when they are issued by DB2 HPU. In most cases, the amount of work that is required to convert Fast Unload JCL to DB2 HPU JCL is significantly reduced.

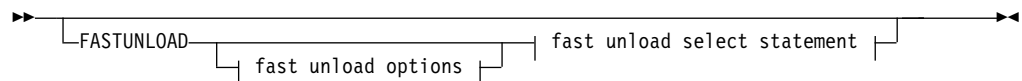
For a description of the syntax shown here, refer to the Fast Unload documentation.

Important:

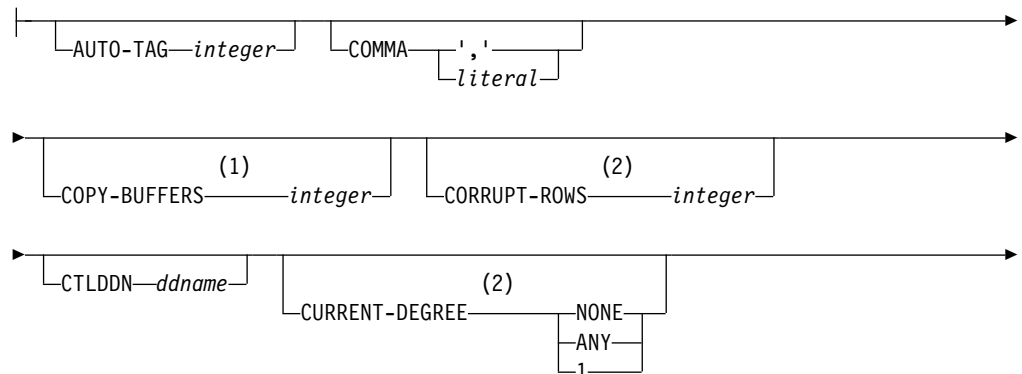
- Do not use this syntax to create new JCL. Fast Unload JCL support is provided only to simplify converting from Fast Unload to DB2 HPU. This syntax is supported to the extent that DB2 HPU can perform processing that is like the processing that is described in the syntax. Many keywords are ignored. Some options are automatically converted to DB2 HPU syntax. For example, OUTPUT-FORMAT LOAD is converted to OUTPUT-FORMAT DSNTIAUL.
- If you specify FASTUNLOAD in SYSIN to use the Fast Unload syntax, precede *subsystem_name/group_attachment_name/job_id* in the PARM field with the EP=UTLGLCTL positional parameter, as shown in the following example:

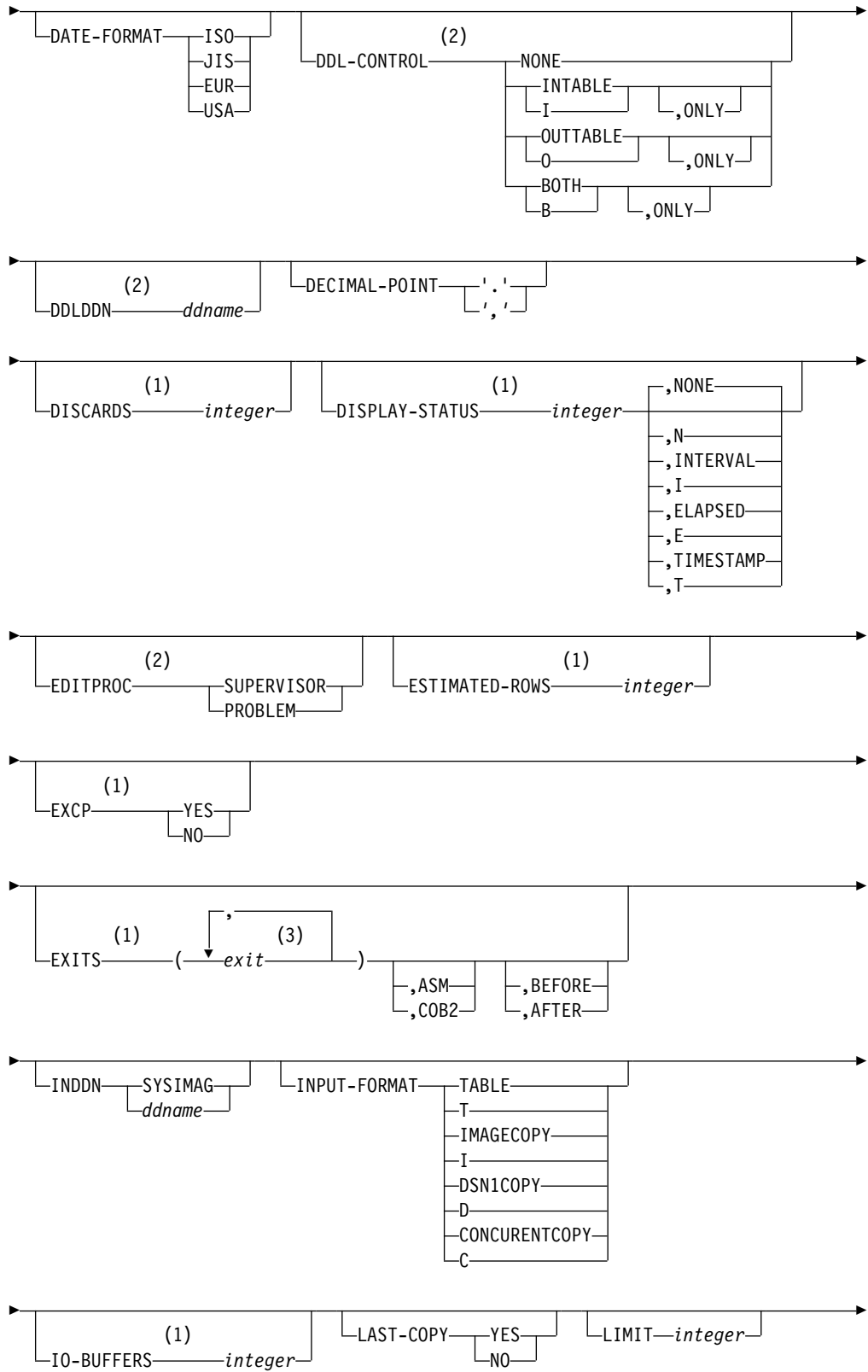
```
PARM='EP=UTLGLCTL/DSN5,DB2UNLOAD'
```

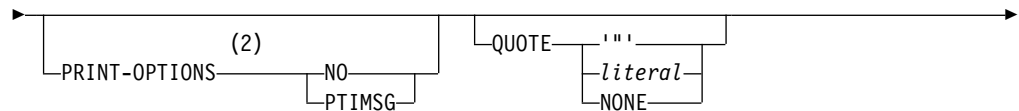
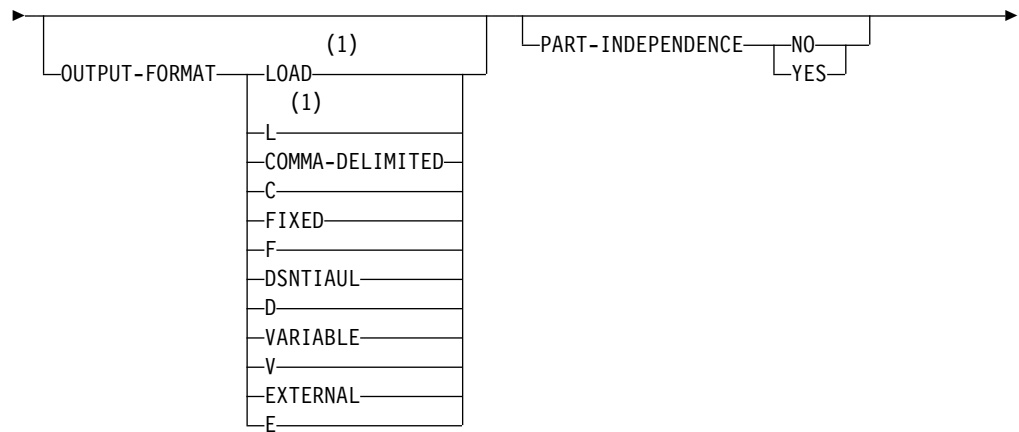
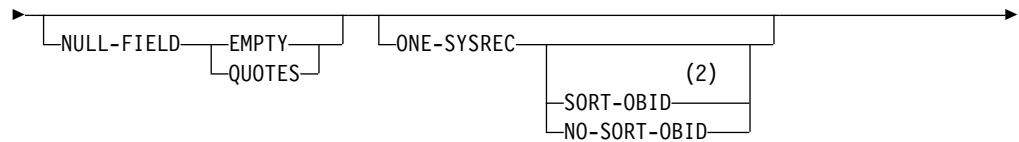
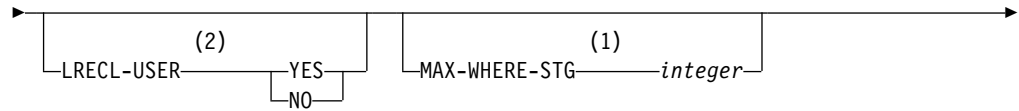
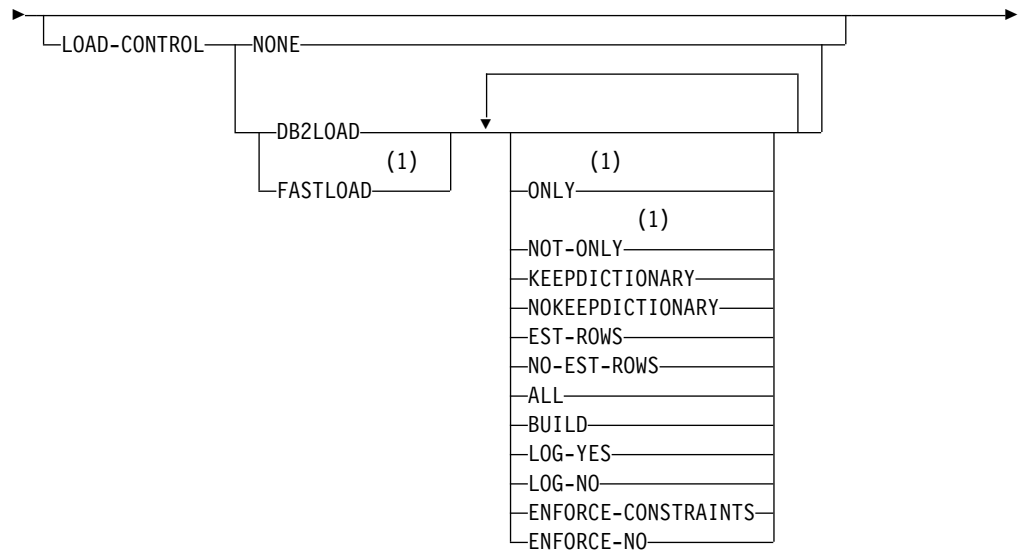
fast unload

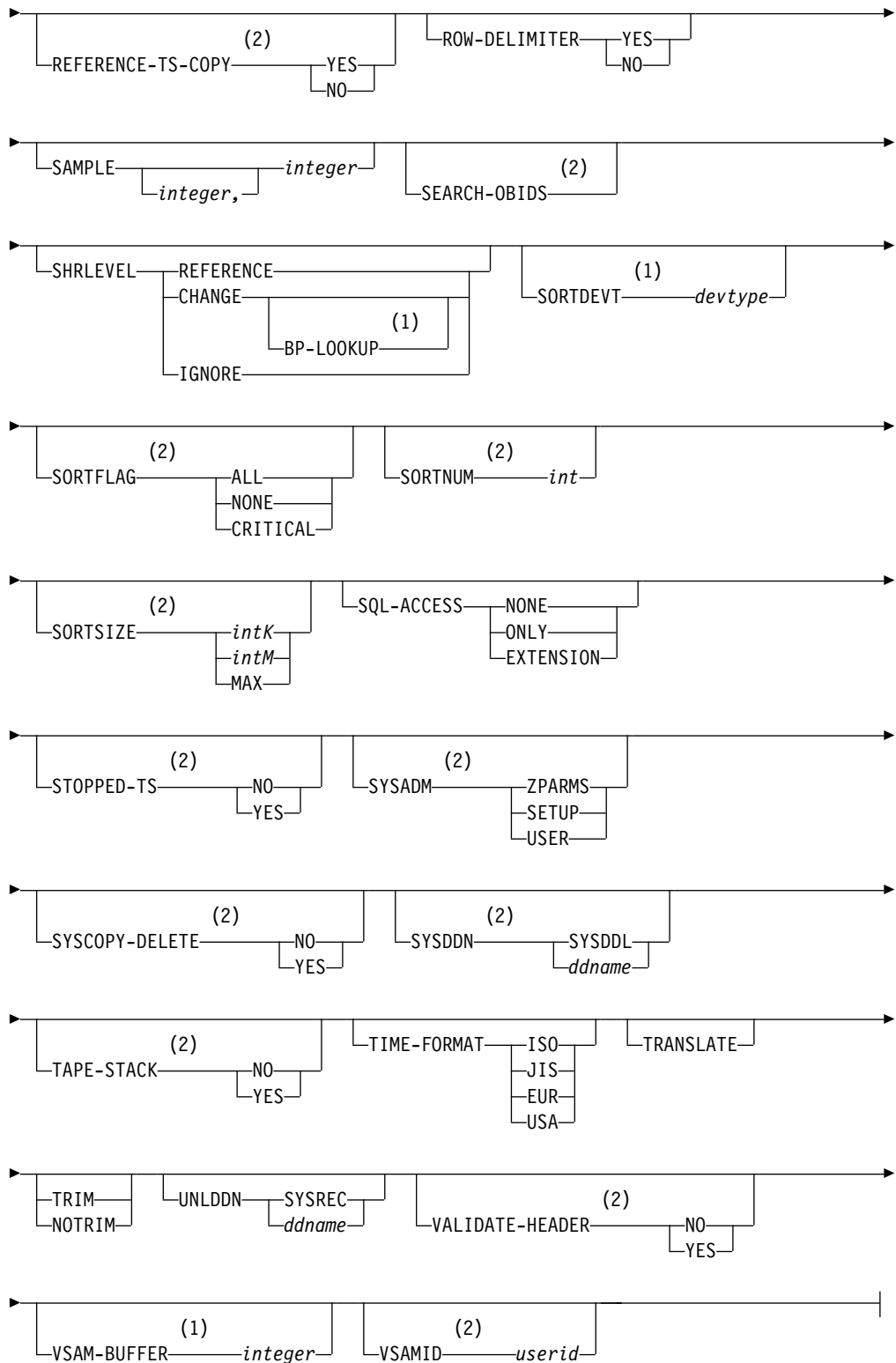


fast unload options:







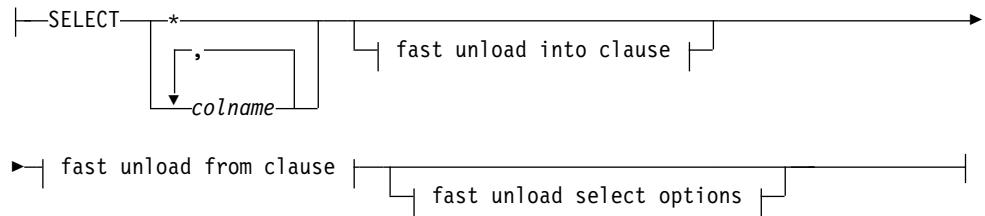


Notes:

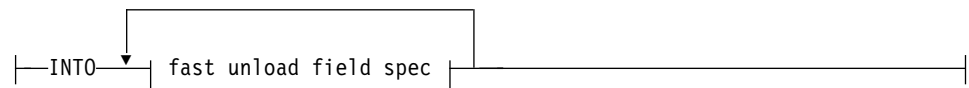
- 1 Keywords that are ignored are identified by an asterisk (*). Dependent keywords and variables are also ignored.
- 2 Keywords that are not supported are identified by two asterisks (**). Dependent keywords and variables are also not supported.

3 You can specify a maximum of four exits.

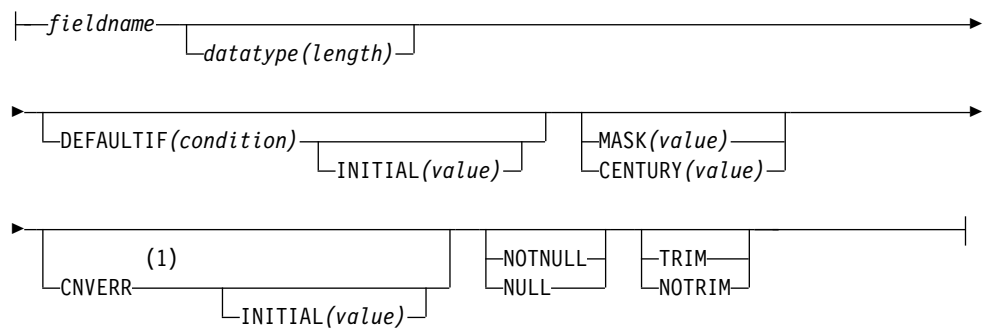
fast unload select statement:



fast unload into clause:



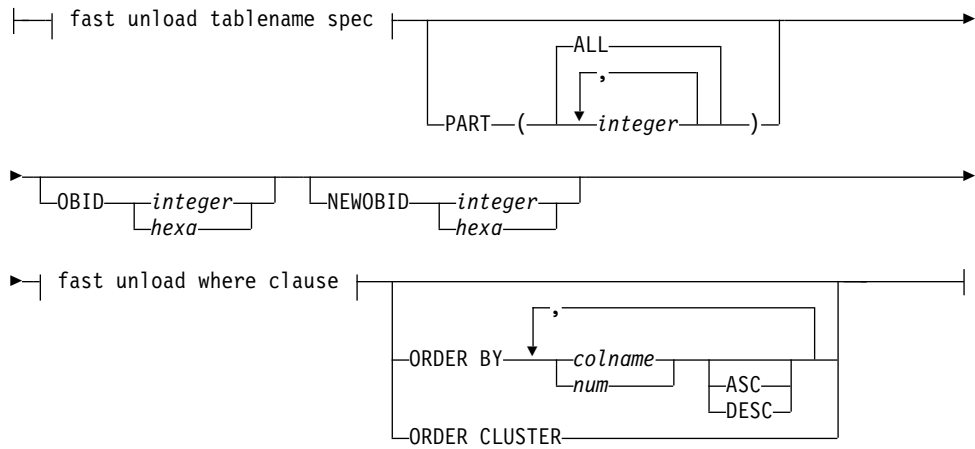
fast unload field spec:



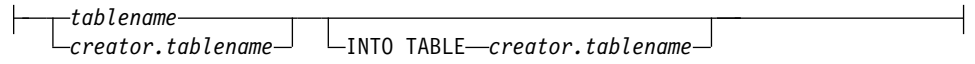
Notes:

- 1 Keywords that are ignored are identified by an asterisk (*). Dependent keywords and variables are also ignored.

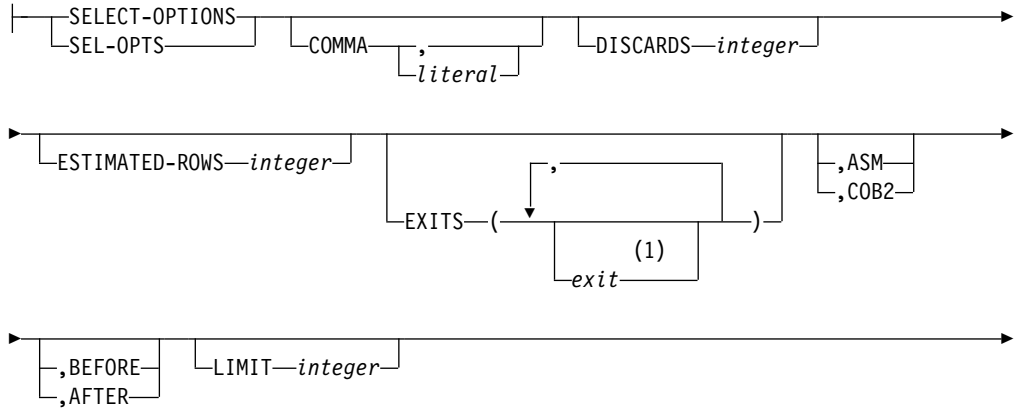
fast unload from clause:

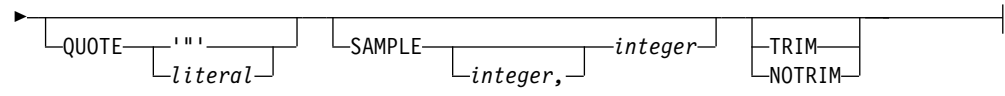
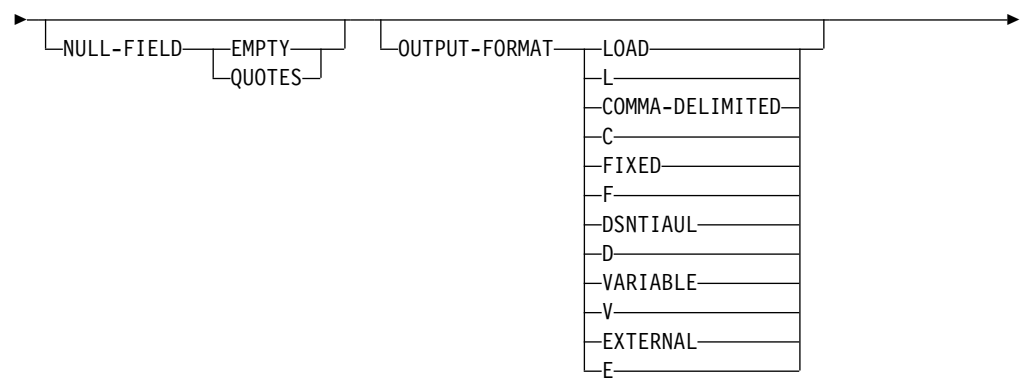
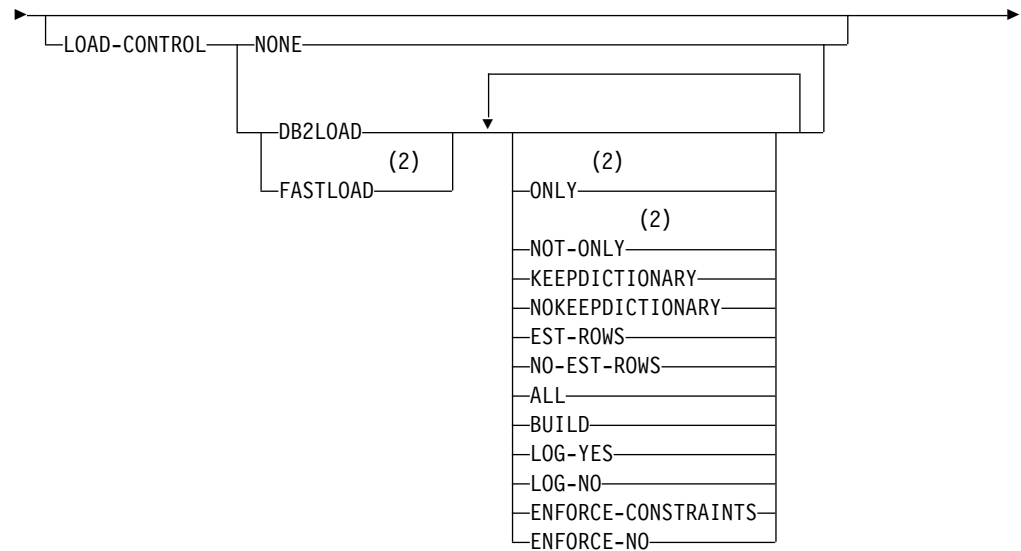


fast unload tablename spec:



fast unload select options:





Notes:

- 1 You can specify a maximum of four exits.
- 2 Keywords that are ignored are identified by an asterisk (*). Dependent keywords and variables are also ignored.

UNLOAD PLUS syntax compatibility

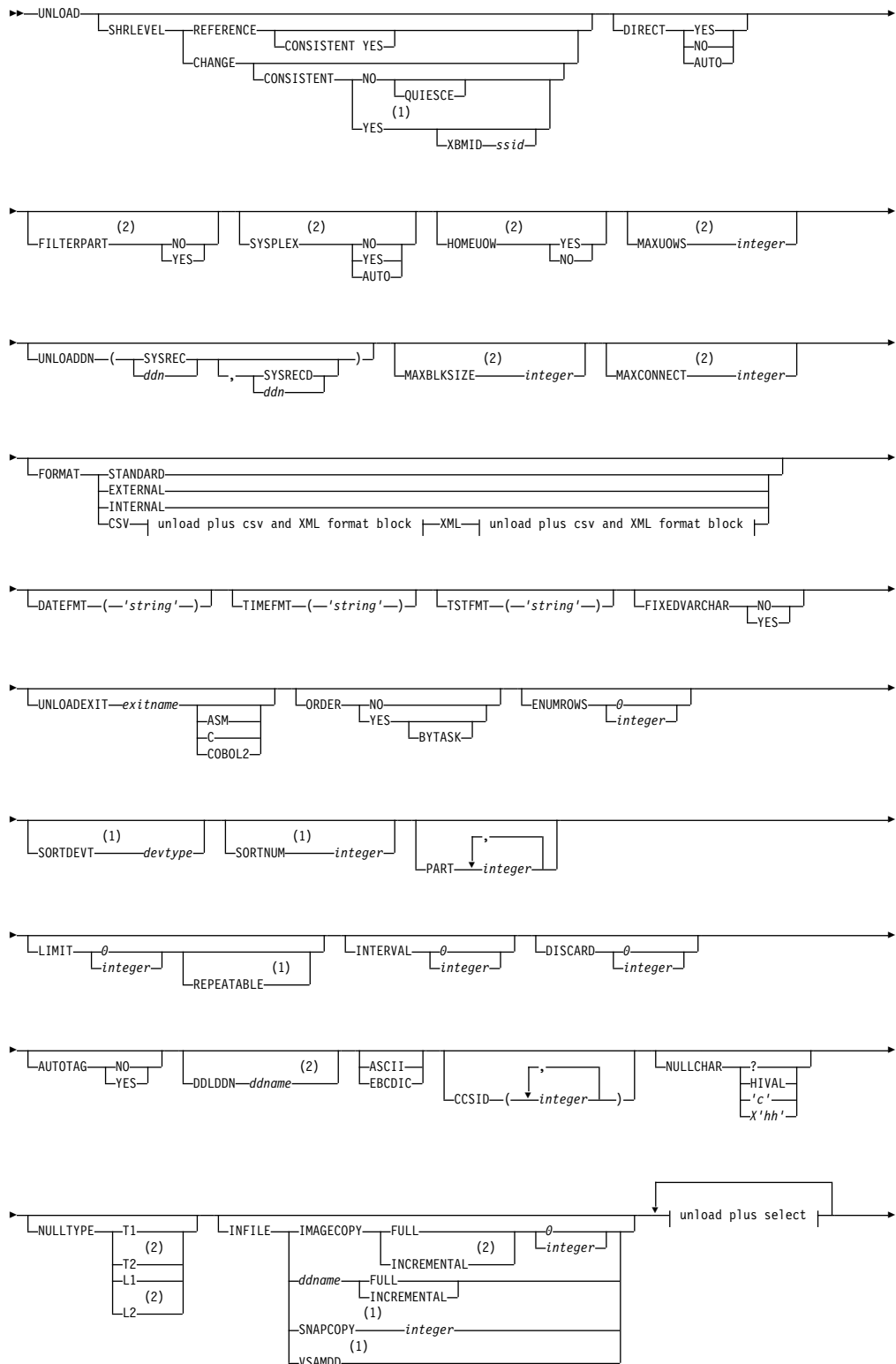
DB2 HPU provides limited support for the JCL that is used with UNLOAD PLUS for DB2, Version 2.1.01; however, some features of UNLOAD PLUS might be ignored or be interpreted differently when issued by DB2 HPU. In most cases, the amount of work that is required to convert UNLOAD PLUS JCL to DB2 HPU JCL is significantly reduced.

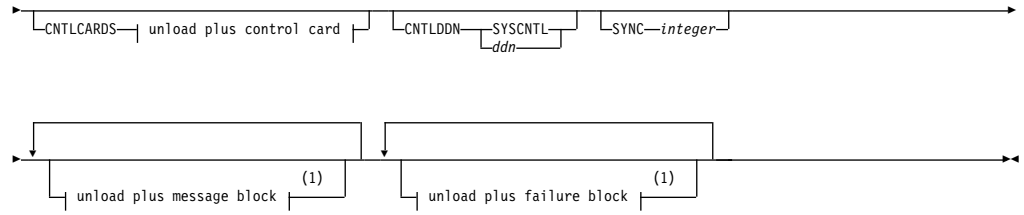
For a description of the syntax shown here, refer to the UNLOAD PLUS for DB2 documentation.

Important: Do not use this syntax to create new JCL. UNLOAD PLUS JCL support is provided only to simplify converting from UNLOAD PLUS to DB2 HPU. This syntax is supported to the extent that DB2 HPU can perform processing that is like

the processing that is described in the syntax. Many keywords are ignored. Some options are automatically converted to DB2 HPU syntax.

UNLOAD syntax

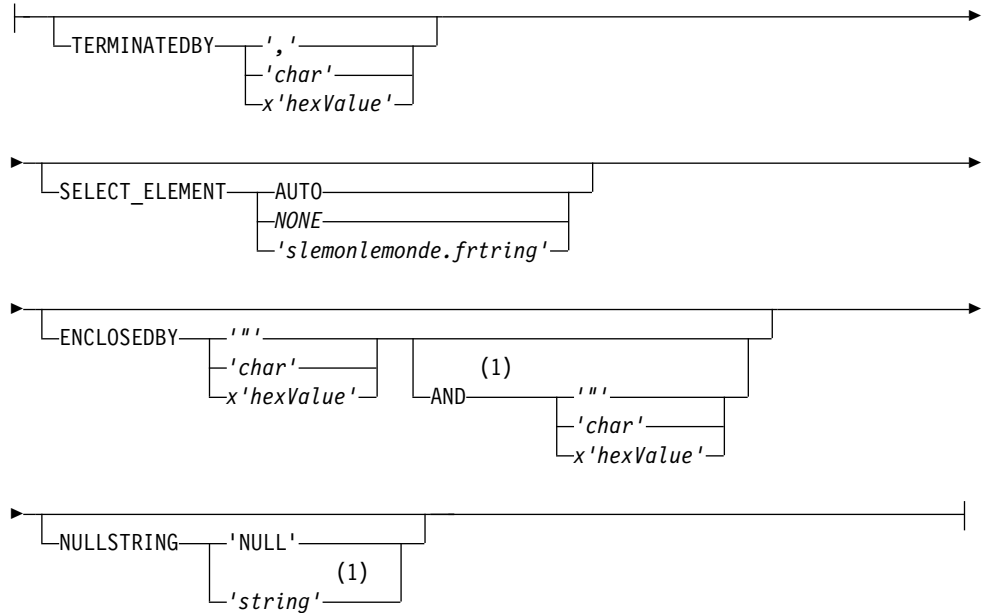




Notes:

- 1 This keyword is ignored. Dependent keywords and variables are also ignored.
- 2 This keyword is not supported. Dependent keywords and variables are also not supported.

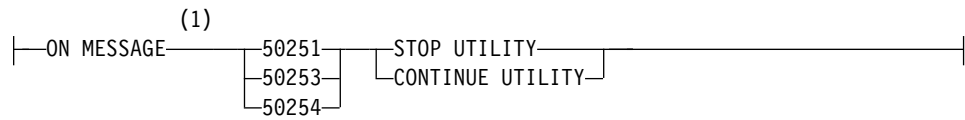
unload plus csv format:



Notes:

- 1 This keyword is not supported. Dependent keywords and variables are also not supported.

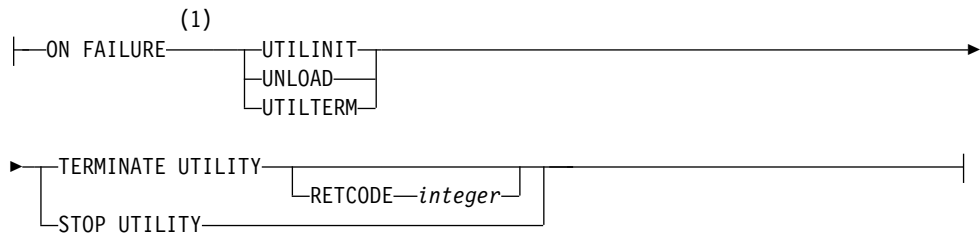
unload plus message block:



Notes:

- 1 This keyword is not supported. Dependent keywords and variables are also not supported.

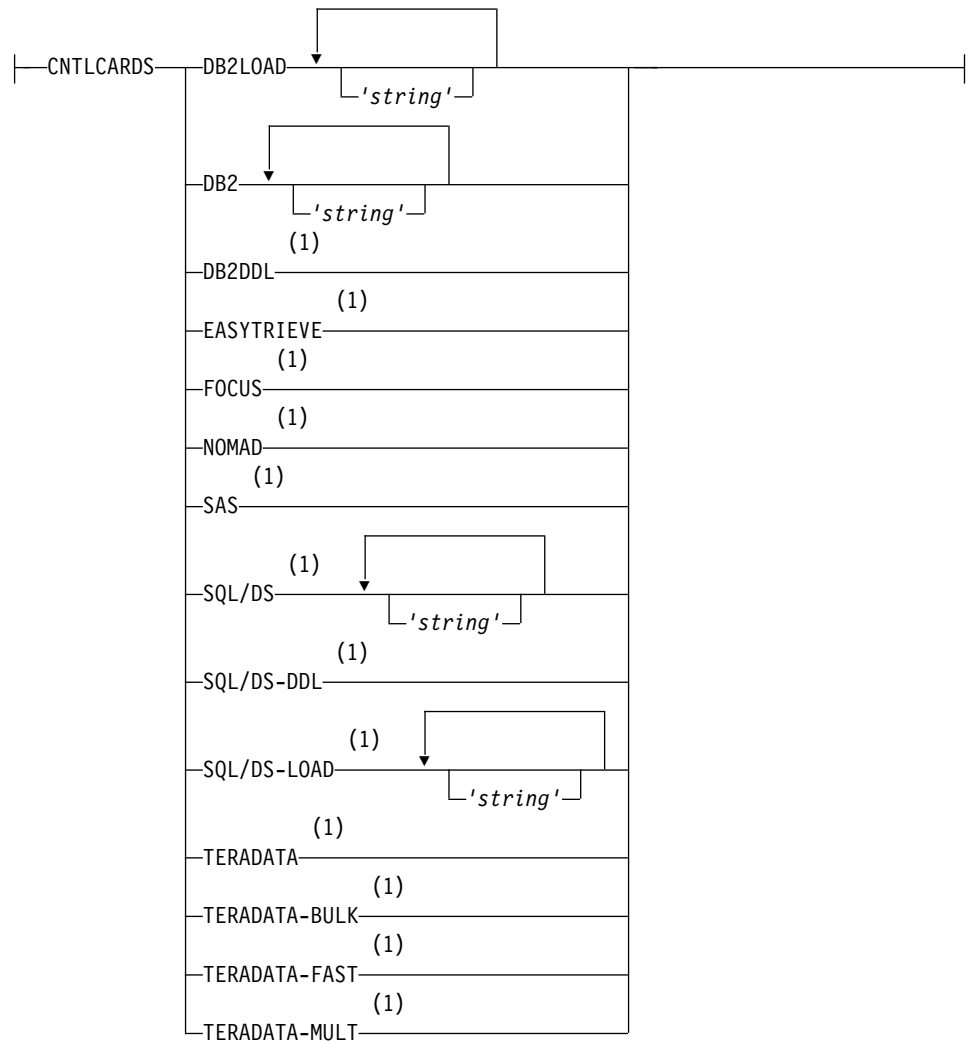
unload plus failure block:



Notes:

- 1 This keyword is not supported. Dependent keywords and variables are also not supported.

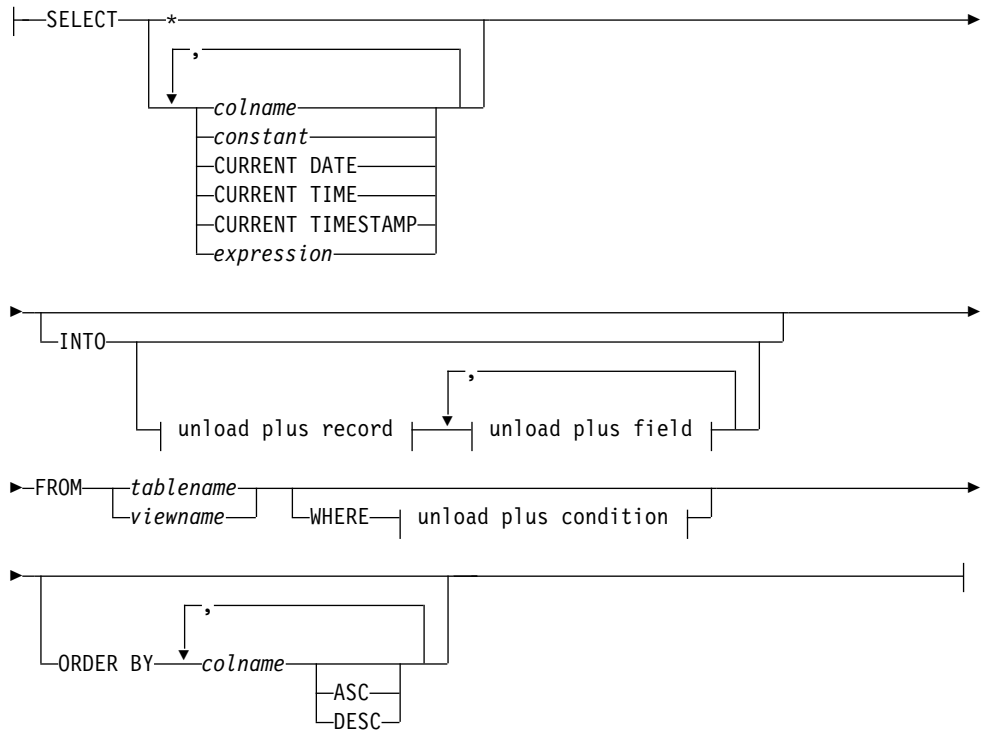
unload plus control card:



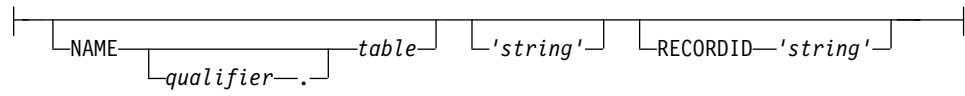
Notes:

- 1 This keyword is ignored. Dependent keywords and variables are also ignored.

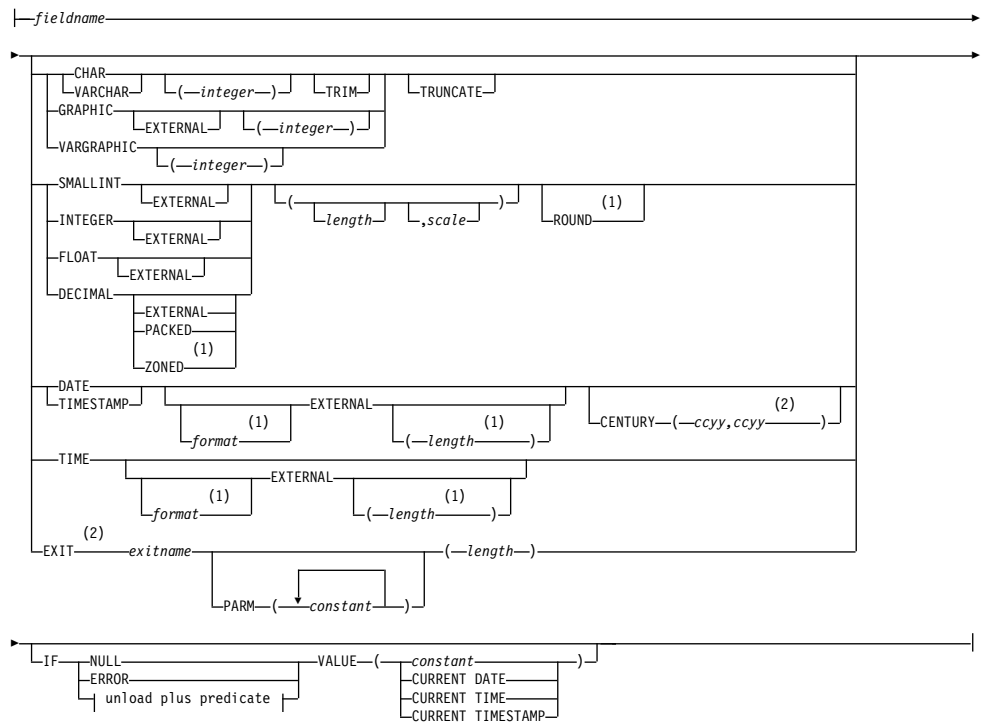
unload plus select:



unload plus record:



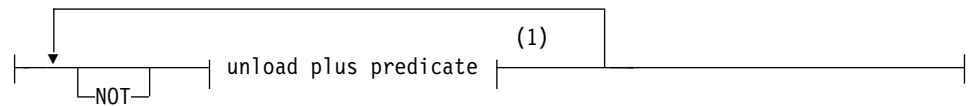
unload plus field:



Notes:

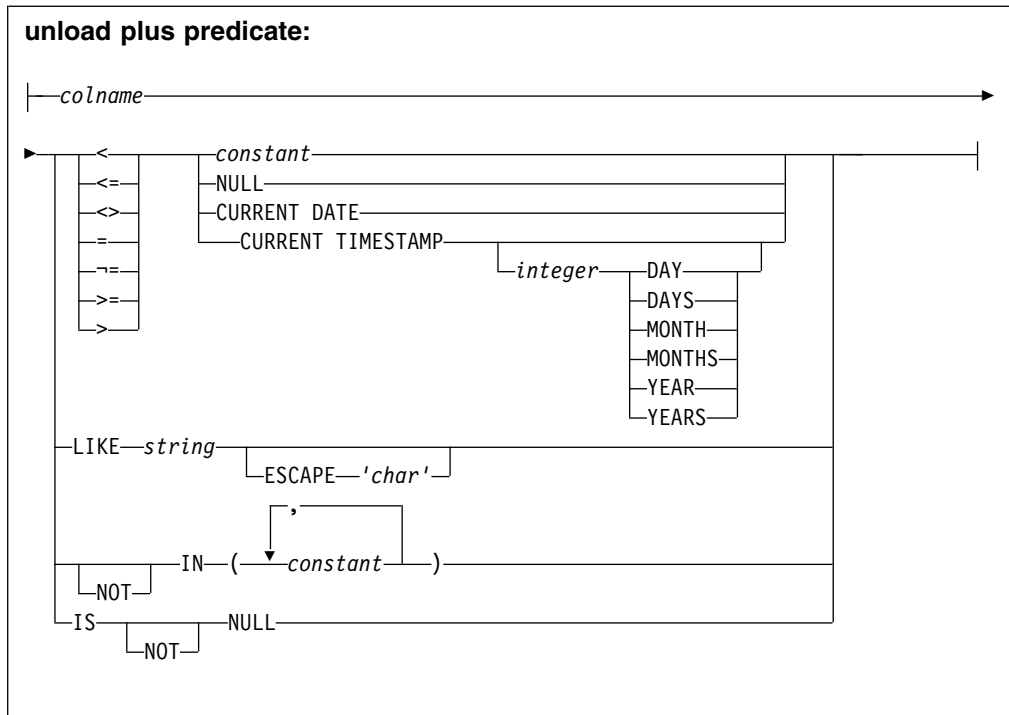
- 1 This keyword is ignored. Dependent keywords and variables are also ignored.
- 2 This keyword is not supported. Dependent keywords and variables are also not supported.

unload plus condition:



Notes:

- 1 Optionally, you can enclose the unload plus predicate in parentheses.



DB2 unload compatibility

DB2 HPU supports the JCL that is used with the DB2 UNLOAD utility for DB2 for z/OS, version 11 and can be operated with a prior DB2 version provided the JCL is still supported by DB2 UNLOAD version 11.

See the related DB2 publication (Utility Guide and Reference) for more details on the DB2 Unload control statements.

However, some features of the DB2 UNLOAD utility might be ignored or be interpreted/implemented differently when performed by DB2 HPU. In most cases, the amount of work that is required to convert a DB2 UNLOAD to DB2 HPU JCL is significantly reduced.

Options

Ignored features:

- NOSUBS: ignored. The message INZU105I is issued. This feature is dealt through VZM008/SCUNSUB.
- DECFLOAT_ROUNDMODE: ignored. The message INZU105I is issued.
- IMPLICIT_TZ: ignored. The message INZU105I is issued.

Altered features:

- MAXERR: applies at the SELECT level rather than at the UNLOAD level.

Restrictions:

- SPANNED YES: DB2 HPU only supports the SPANNED YES option for LOB data.

Field specification

Restriction:

- POSITION: only POSITION(*) is supported. End with completion code 8 and a message INZU052E is issued.

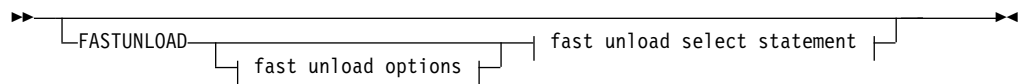
Not supported:

- timestamp with time zone-spec: not supported. End with completion code 8 and a message INZU052E is issued.

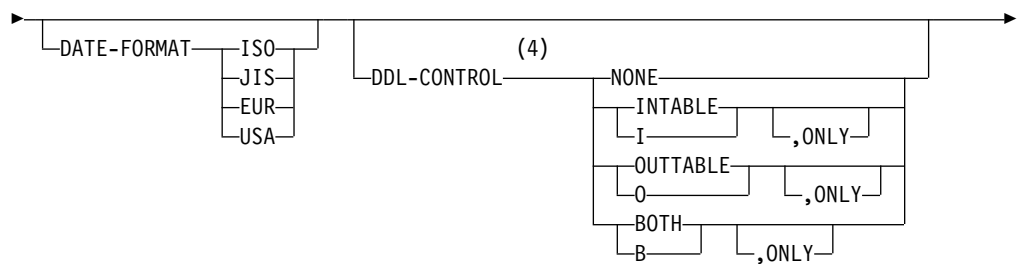
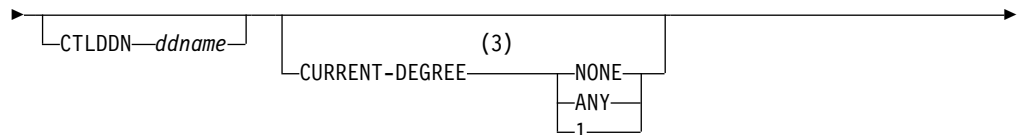
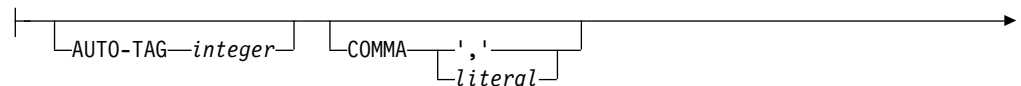
Important:

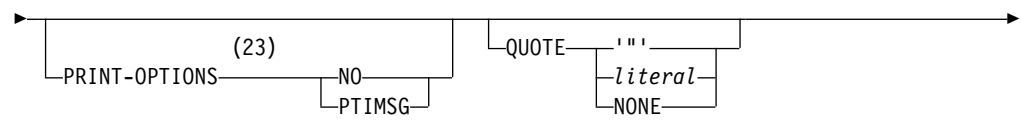
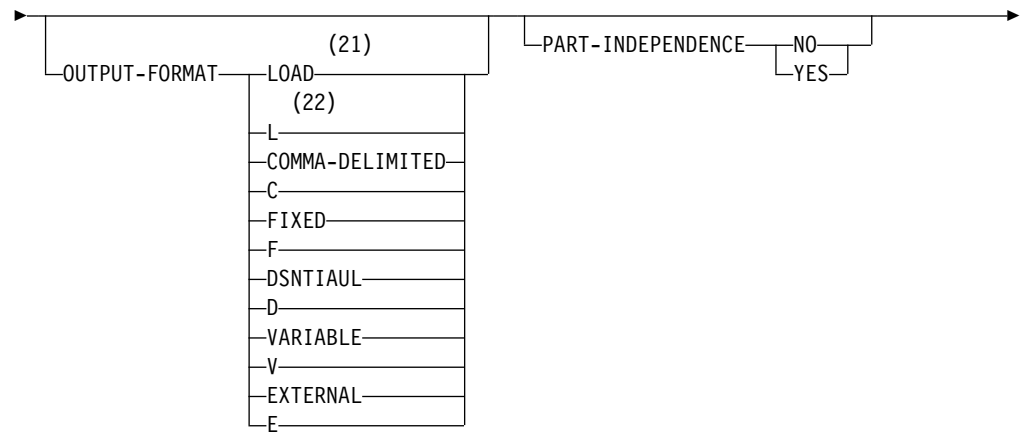
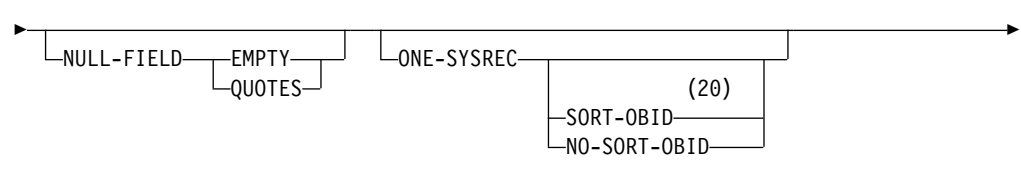
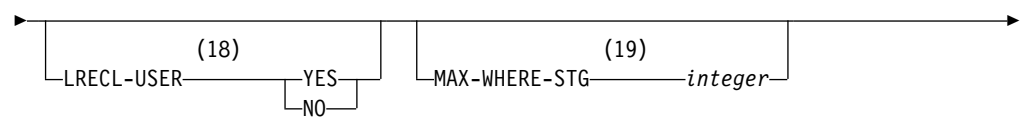
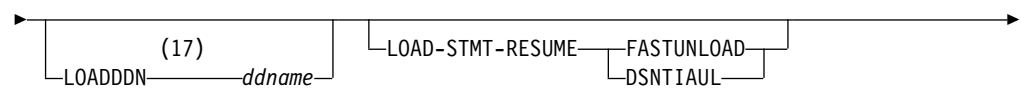
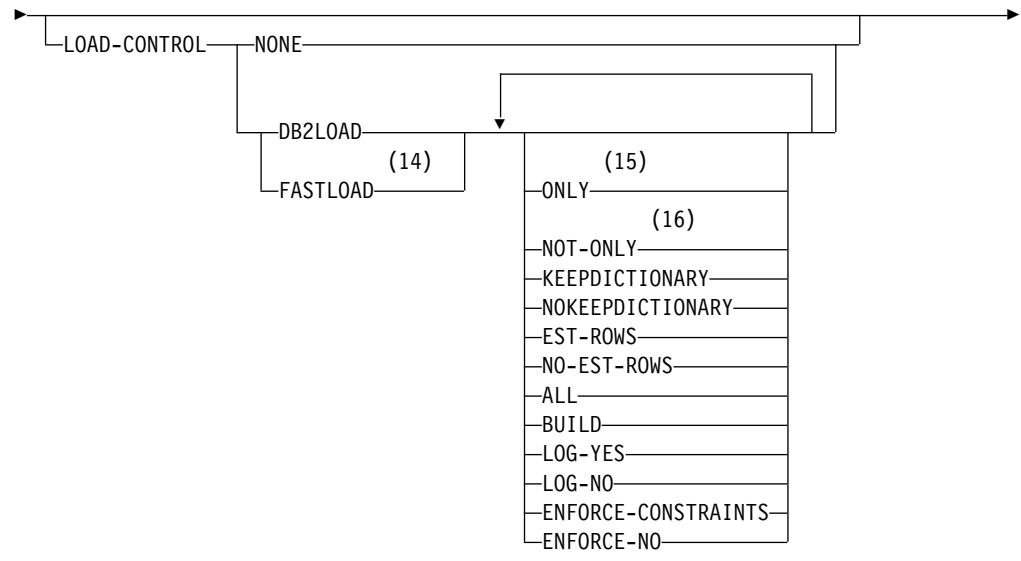
Do not use this syntax to create new DB2 HPU JC. The D2 UNLOAD support is provided only to simplify converting from DB2 UNLOAD to DB2 HPU. This syntax is supported to the extent that DB2 HPU can perform processing that is like the processing that is described in the syntax. Several keywords are ignored. Some options are automatically converted to DB2 HPU features.

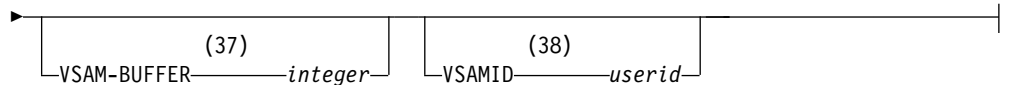
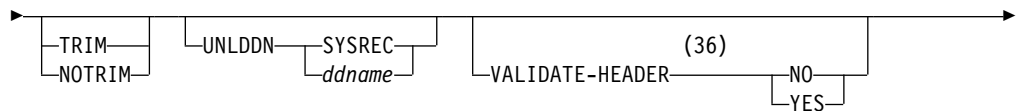
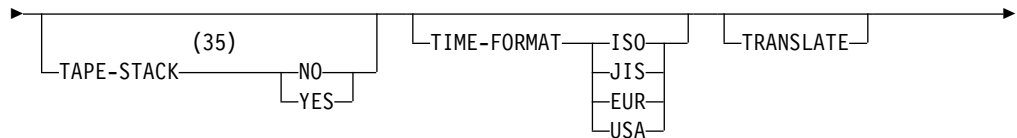
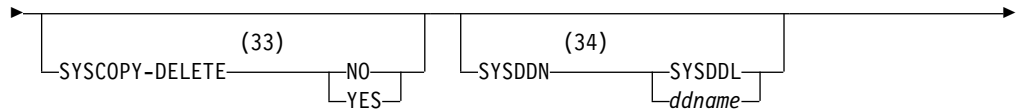
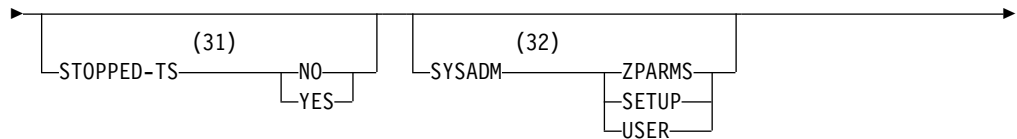
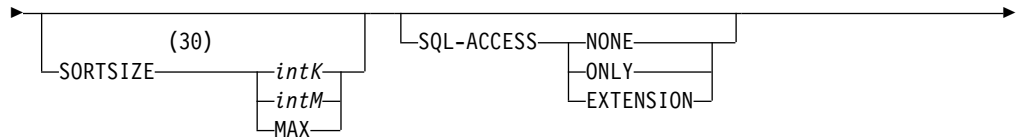
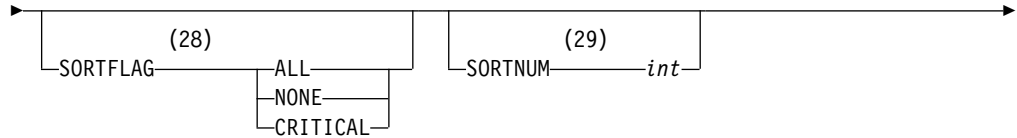
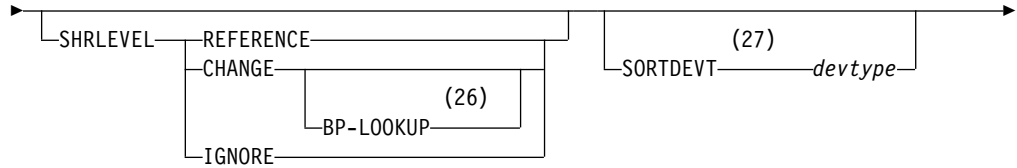
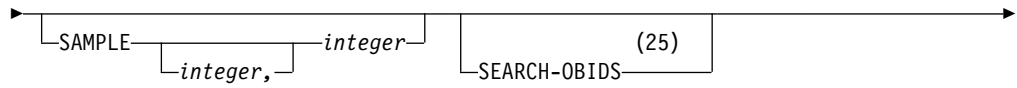
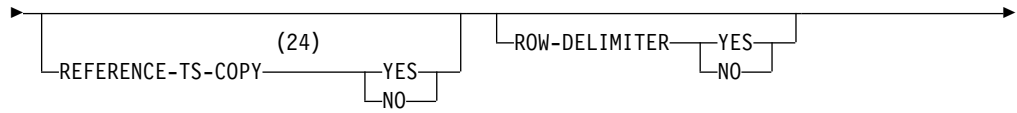
fast unload



fast unload options:







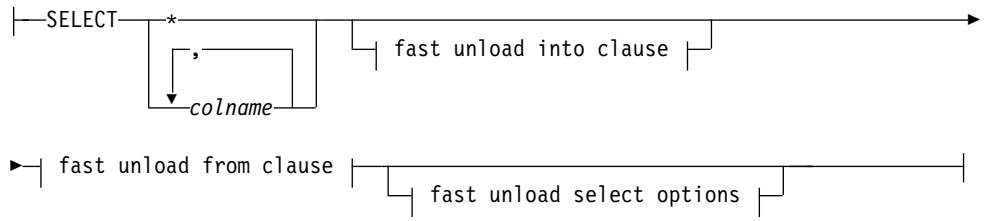
Notes:

- 1
- 2
- 3

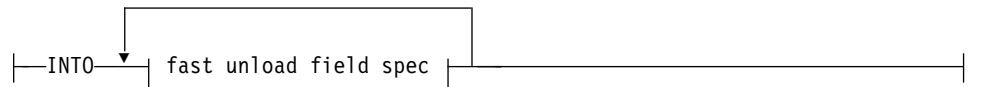
	4
	5
	6
	7
	8
	9
	10
	11
	12
	13
	14
	15
	16
	17
	18
	19
	20
	21
	22
	23
	24
	25
	26
	27
	28
	29
	30
	31
	32
	33
	34
	35
	36
	37
	38

You can specify a maximum of four exits.

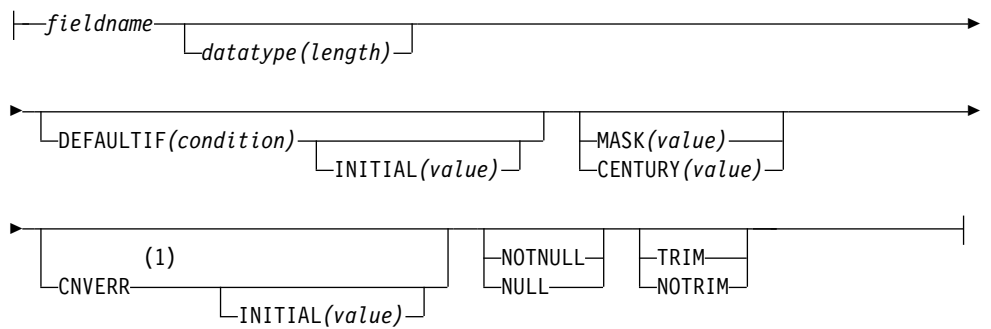
fast unload select statement:



fast unload into clause:



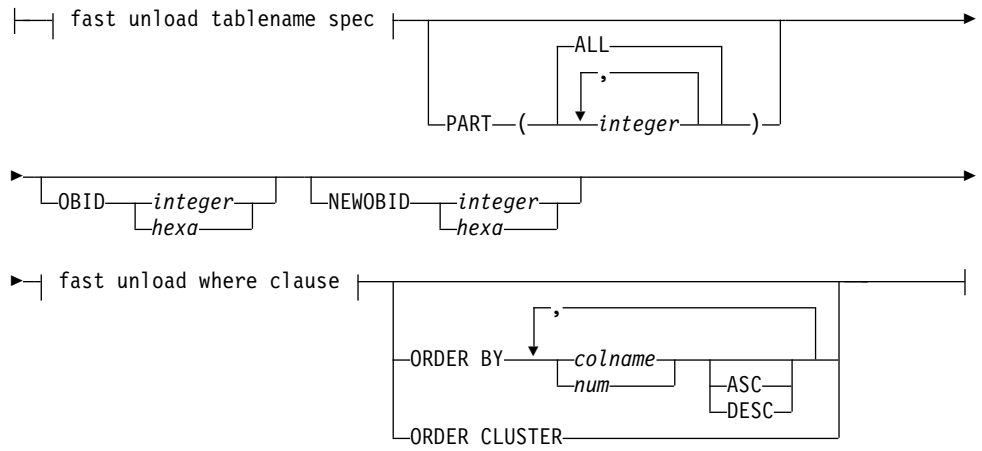
fast unload field spec:



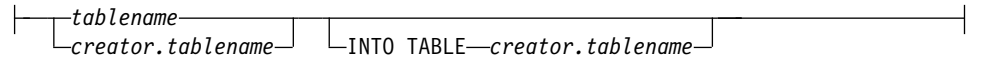
Notes:

1

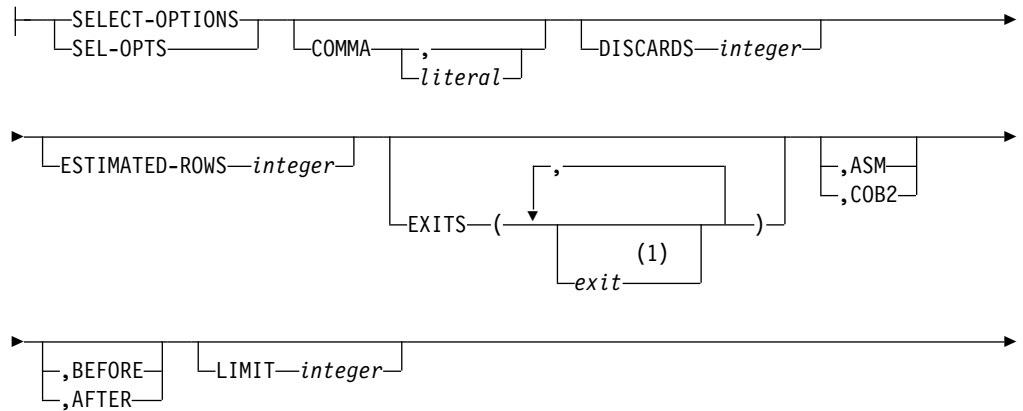
fast unload from clause:

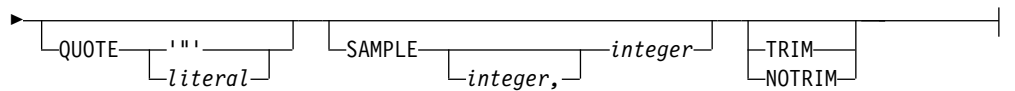
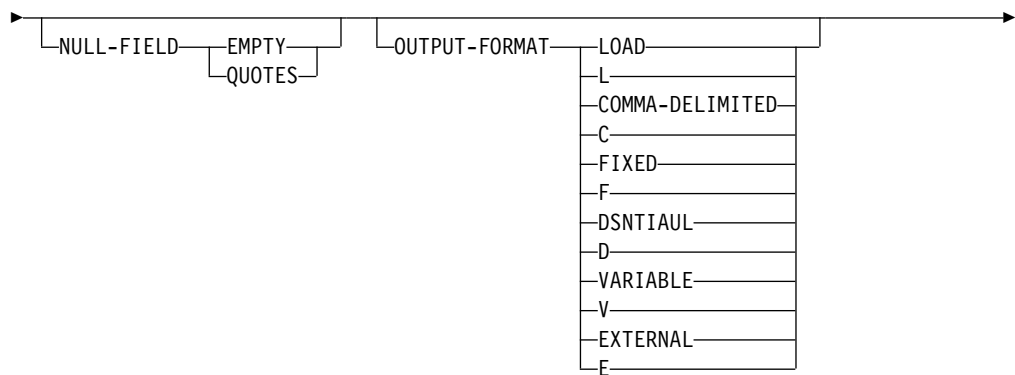
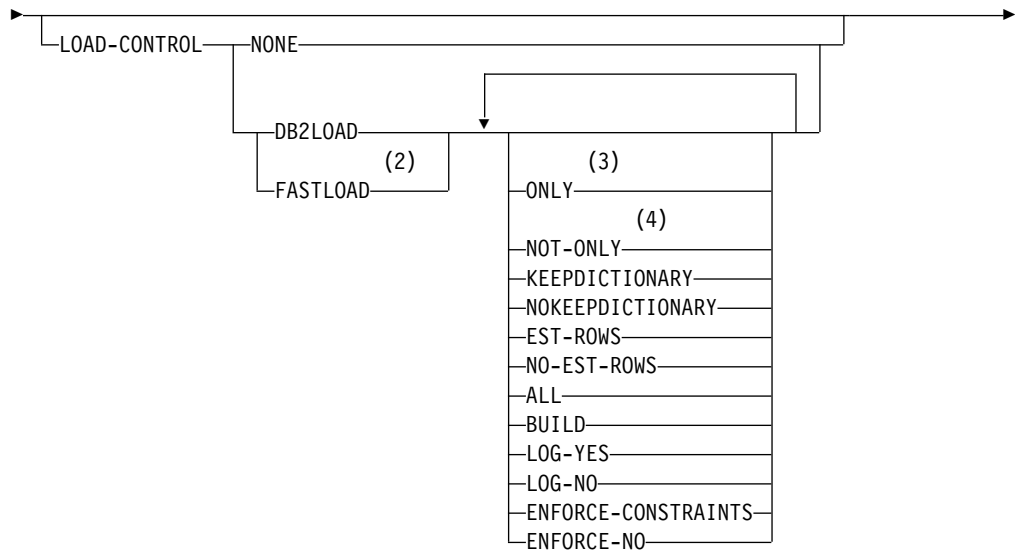


fast unload tablename spec:



fast unload select options:





Notes:

- 1 You can specify a maximum of four exits.
- 2
- 3
- 4

Example: Usage of a DB2 UNLOAD JCL with the HPU load module

In this example, you can unload in XML format from table column with the additional AUTOTAG field.

Specify PGM=INZUTILC as in the example below as creating an alias for the DSNUTILB load module might not be relevant since DSNUTILB is used for other utilities than the UNLOAD.

```

//UNLOAD EXEC PGM=INZUTILC,PARM='DB81,DB2UNLOAD',
//REGION,=0M,TIME=1440
//STEPLIB DD DSN=DB2UNL.SINZLINK,DISP=SHR
//DD DSN=DSN810.SDSNLOAD,DISP=SHR

```

```
| //SYSPRINT DD SYSOUT=*
| //SYSREC DD SYSOUT=*
| //SYSUDUMP DD SYSOUT=*
| //SYSPUNCH DD SYSOUT=*
| //SYSIN DD*
| UNLOAD TABLESPACE DSN8D10A.DSN8S10E
| FROM TABLE DSN81010.EMP
|
```

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 Director of Licensing
IBM Corporation
North Castle Drive
MD-NC119
Armonk, NY 10504-1785
US.

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 is for planning purposes only. The information herein is subject to change before the products described become available.

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.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Index

Special characters

397

∅ keyword 89

&variable keyword 89

A

abend codes 329

accessibility 10

overview 6

ACCPREP parameter 447

ACTLGUSR parameter 397

Admin Tool

starting DB2 HPU interactive

component 215

APF LOAD modules library 395

APF LOAD modules parameter 395

ASCII keyword 168

ASIS keyword 168

authority privileges 75

authorization 13

AUTOTAG keyword 134

B

basic predicate 174

batch utility

enabling 13

generating jobs 14, 213

overview 75

benefits 2

between predicate 174

BLOBF parameter 134

block syntax 82

buffering technique 8

BUFNO keyword 89

C

CCSID keyword 168

CCSID parameter 134

changing display options 29

character expression keyword 89

CLONE keyword 110

CLONED keyword 84

codes

abend 329

parameter block 257

COL keyword 198

COLLOBJT parameter 397

COLUMNS primary command 213

compatibility

DB2 unload 476

DB2 UNLOAD utility 75

fast unload syntax 463

overview 7

UNLOAD PLUS syntax 469

Complete table name panel 233

complex processing 110

components 3

configuration

authorizing 13

enabling batch utility 13

Confirm to leave the current Unload

panel 218

consistency

output data 69

unloading 71

conversions

DATE format 134

supported output data types 462

TIME format 134

TIMESTAMP format 134

cookie policy 487

Copy DB2 Entries panel 50

COPYDDN

specifying file parameters 222

COPYDDN File Parameters panel 222

COPYDDN keyword 110

using with TEMPLATE

statements 102

COPYDDN primary command 213

CTRLLIBM parameter 411

customization

associated list

adding DB2 entries 37

overview 15

associating DB2 entries 37

browsing parameters 50

changing display options 29

changing parameters 31

component 15

copying DB2 entries 50

Create a DB2 Entry panel 37

creating DB2 entries 37

customization jobs

deleting 53

displaying 52

generating 46

maintaining 53

regenerating 46

renaming 53

sort sequence 47

submitting 47, 52

customization library

deleting jobs 53

maintaining 53

overview 18

recustomizing 53

renaming jobs 53

customization library qualifier

specifying 25

Customized status 15

Customizer Workplace panel 46

customizing a new version of a

product 31

customizing a product for the first

time 31

customizing settings 25

customization (*continued*)

data sets

customization library 18

data store 18, 20

DASTASTOR 20

Discover EXEC library 18

metadata library 18, 20

data store

overview 18

data store data set

specifying 25

DB2 Administration Tool

parameters 438

DB2 data sharing members

adding 37

associating 37

copying 50

creating 37

DB2 entries 15

adding 37

associating 37

copying 50

creating 37

defining 46

deleting 51, 52

generating jobs for 46

removing 51

selecting 46

specifying 46

unassociating 51, 52

DB2 group attach field

specifying 25

DB2 group attach names

adding 37

associating 37

copying 50

creating 37

DB2 HPU conversion parameters 439

DB2 HPU DB2 parameters 397

DB2 HPU file management

parameters 440

DB2 HPU library dsnames 395

DB2 HPU output data parameters

output data 411

DB2 HPU sort parameters 453

DB2 HPU tuning parameters 447

DB2 parameters

defining 44

editing 44

DB2 Parameters panel 44

DB2 subsystems

adding 37

associating 37

copying 50

creating 37

defining DB2 parameters 44

defining LPAR parameters 41

defining parameters 39, 46

defining product parameters 39

deleting DB2 entries 52

deleting jobs 32

- customization (*continued*)
 - Discover Customized Product Information panel 35
 - Discover EXEC
 - customizing a new version of a product 31
 - overview 18
 - retrieving product information automatically 35
 - Discover process parameters 395
 - Discovered status 15
 - discovering product information 35
 - display options 29
 - displaying jobs 52
 - displaying panel text 29
 - editing LPAR parameters 41
 - editing parameters 31
 - editing product parameters 39
 - Errors in Customization status 15
 - finding trace data set 267
 - Finish Product Customization panel 47
 - first-time 31
 - first-time customization 31
 - generating jobs 46
 - hiding panel text 29
 - high-level qualifier 15
 - Incomplete status 15
 - integrating DB2 HPU into DB2 Admin 55, 438
 - integrating DB2 HPU into DB2 Launchpad 54
 - job sort order 47
 - jobs
 - deleting 53
 - displaying 52
 - maintaining 53
 - renaming 53
 - sort order 47
 - submitting 47, 52
 - LPAR parameters
 - defining 41
 - editing 41
 - LPAR Parameters panel 41
 - LPARs 53
 - maintaining jobs 53
 - master list
 - adding DB2 entries 37
 - Associate DB2 Entry for Product panel 37
 - overview 15
 - maximizing information panels 29
 - metadata libraries
 - specifying 33
 - metadata library
 - maintenance, best practices 20
 - overview 18
 - specifying 25
 - modifying parameters 31
 - modifying settings 25
 - multiple instances 25
 - multiple-LPAR environment 53
 - Not Required status 15
 - options 29
 - overview 13
 - panel display options 29

- customization (*continued*)
 - panels
 - Associate DB2 Entry for Product 37
 - Create a DB2 Entry 37
 - Customizer Workplace 46
 - DB2 Parameters 44
 - Discover Customized Product Information 35
 - Finish Product Customization 47
 - LPAR Parameters 41
 - Product Parameters 39
 - Specify the Metadata Library 33
 - parameters
 - browsing 50
 - defining 39, 46
 - viewing 50
 - preparing to use Tools
 - Customizer 25
 - prerequisites 13
 - product 15
 - product parameters
 - changing 32
 - defining 39
 - editing 32, 39
 - modifying 32
 - Product Parameters panel 39
 - Ready to Customize status 15
 - recustomization 31, 32
 - recustomizing 32
 - recustomizing a product 31
 - removing DB2 entries 51
 - roadmaps 31
 - customizing for the first time 31
 - first-time customization 31
 - reustomizing 32
 - Specify the Metadata Library panel 33
 - specifying data sets 25
 - specifying metadata libraries 33
 - starting Tools Customizer 24
 - status types
 - Customized 15
 - Discovered 15
 - Errors in Customization 15
 - Incomplete 15
 - Not Required 15
 - Ready to Customize 15
 - steps 13
 - submitting jobs 47
 - terminology 15
 - trace data set 267
 - troubleshooting 267
 - finding trace data set 267
 - user job card settings
 - specifying 25
 - viewing parameters 50
 - customization library
 - overview 18
 - customization library qualifier
 - specifying 25
 - customizing settings 25

D

- data consistency
 - unloading 71

- data conversion
 - DATE format 134
 - TIME format 134
 - TIMESTAMP format 134
- data sets
 - output 168
- data sources 57
 - full image copies (FIC) 59
 - incremental image copies 59
 - last full image copies 60
 - nonpartitioned table spaces 57
 - partitioned table spaces 57
- data store
 - maintenance, best practices 20
 - overview 18
- data store data set
 - specifying 25
- data types
 - nonnumeric 459
 - numeric 458
- Database and Tablespace Selection panel 217
- DATABASE keyword 84
- DATACLAS keyword 89
- DATE format types
 - specifying output data 459
- DATE keyword 134
- DATEDDEL parameter 411
- DATEDELIM keyword 134
- DB2 Admin
 - integrating DB2 HPU 55
- DB2 Admin Tool
 - starting DB2 HPU interactive component 215
- DB2 Administration Tool
 - integrating with DB2 HPU 456
- DB2 Administration Tool parameters
 - customizing 438
- DB2 catalog
 - access 69
 - Direct 69
 - SQL 69
- DB2 group attach field
 - specifying 25
- DB2 HPU conversion parameters
 - customizing 439
- DB2 HPU DB2 parameters
 - customizing 397
- DB2 HPU file management parameters
 - customizing 440
- DB2 HPU library dsnames
 - customizing 395
- DB2 HPU output data parameters
 - customizing 411
- DB2 HPU sort parameters
 - customizing 453
- DB2 HPU tuning parameters
 - customizing 447
- DB2 keyword 110
- DB2 Launchpad
 - integrating DB2 HPU 54
 - starting DB2 HPU interactive component 217
- DB2 parameter 218
- DB2 QUIESCE authority 75
- DB2 SELECT privilege 75

- DB2 subsystems
 - specifying 217
- DB2 System panel 217
- DB2 Tools Launchpad panel 217
- DB2 UNLOAD utility compatibility 75
- DB2DBRM 395
- DB2PARAL parameter 397
- DBCLOBF parameter 134
- DBRM library 395
- DBRM parameter 395
- DDLDDN keyword 110
 - using with TEMPLATE statements 102
- ddnames
 - allocating by DB2 HPU
 - ICxx-ICxxxxnn 79
 - LBFnnnnn 79
 - SORT 79
 - allocating by user
 - copydd 80
 - INFLIB 80
 - INZPLIB 80
 - JOBLIB 80
 - loaddd 80
 - outdd 80
 - STEPLIB 80
 - SYSABEND 80
 - SYSIN 80
 - SYSPRINT 80
 - SYSTEM 80
 - unldd 80
 - UTPRINT 80
- decreasing memory 265
- DELIM parameter 134
- DELIMITED block syntax 195
- DELIMITED format
 - description 2
 - examples 63
 - overview 195
- DELIMITED keyword 190, 195
- DFSIGDCB parameter 411
- diagnostic information
 - gathering 267
- DIR keyword 89
- disability 10
- Discover EXEC
 - overview 18
- Discover process parameters
 - customizing 395
- DISP keyword 89
- display options 29
- displaying panel text 29
- DISPLUSR parameter 397
- documentation
 - accessing 5, 10
 - sending feedback 5
- documentation changes 1
- DSN keyword 89
- DSN1 Table Spaces panel 215
- DSN1 Tables, Views, and Aliases
 - panel 215
- dsnames
 - installation libraries 395
- DSNTIAUL block syntax 193
- DSNTIAUL format
 - creating sequential data sets 201
 - description 2

- DSNTIAUL format (*continued*)
 - examples 63, 205
 - overview 193
- DSNTIAUL keyword 190, 193
- DSNTYPE keyword 89
- DTDELIM parameter 411
- DTNULDLM parameter 411
- DYNALLOC macro (SVC 99) 89

E

- EBCDIC keyword 168
- Edit Expression panel 238
- EDIT, SUBmit, SaveJCL, JobCard
 - panel 249
- encoding schemes 65
- ENFORCE COLUMN ORDER
 - parameter 134
- example
 - Generating the access path during the data extraction using CURRENT OPTIMIZATION HINT 168
 - Requesting the input data integrity to be checked with the LEVEL1 controls 134
 - Unloading in XML format using variables 164
 - Unloading into a data set allocated via a TEMPLATE with DSNTYPE EXTREQ 105
- examples
 - Assembler user exit 259
 - calling user exit 67
 - changing column names 205
 - changing columns 64
 - changing data types 205
 - COBOL/2 user exit 261
 - committed data 108
 - concatenating columns 64
 - consistent image copies 60
 - creating DB2 LOAD-compatible sequential data sets 202
 - creating sequential data sets
 - DSNTIAUL format 201
 - USER format 200
 - DELIMITED format 63
 - DSNTIAUL format 63
 - EXTERNAL format 65
 - FORMAT INTERNAL 190
 - full image copies (FIC) 59
 - HFS files 104, 207
 - hierarchical file system (HFS) 207
 - incremental image copies 59
 - INTERNAL_FORMAT 110
 - INZDB21X member 456
 - last full image copies 60
 - last image copies 132
 - limiting unloaded rows 67
 - LOB data 207
 - logical unload 62
 - MAX_EXPECTED_ROWS 209
 - maximizing resources 209
 - nonpartitioned table spaces 57
 - OUTFREQROWS keyword 67
 - OUTMAXROWS keyword 67
 - padding columns 64
 - partitioned table spaces 57

- examples (*continued*)
 - period specification clauses 207
 - physical unload 61
 - processing multiple UNLOAD and SELECT statements 203
 - renaming columns 205
 - Requesting progress messages via the UNLOAD command 133
 - running two SELECT statements 76
 - sampling on unloaded rows 61
 - SELECT statement 200
 - sort operations 209
 - sort utilities 108
 - sorting records 65
 - SORTUTIL keyword 108
 - specifying one output file per partition 66
 - SQLPART 174
 - stacking on tapes report 209
 - SYSPRINT 209
 - temporal tables 207
 - timestamp precision 208
 - Unloading a table space containing 3 tables and requesting the NUMRECS option to be specified in the generated LOAD statement using the actual number of unloaded rows 167
 - unloading by using a delimiter 63
 - unloading by using a separator 63
 - unloading data by using a TEMPLATE statement 103, 104
 - Unloading data from a table with informational header 165, 190
 - unloading files in FORMAT EXTERNAL 65
 - unloading from a consistent image copies 60
 - unloading from last full image copies 60
 - unloading full image copies (FIC) 59, 63
 - unloading incremental image copies 59
 - unloading LOB columns 204
 - unloading LOB data 207
 - Unloading LOB data into spanned format 165
 - unloading LOBs 104
 - unloading nonpartitioned table spaces 57
 - unloading partitioned table spaces 57
 - unloading partitions into separate output files 67
 - unloading tables 130
 - unloading with COPYDDN 59
 - unloading with DDLDDN 59
 - unloading with FORMAT INTERNAL 190
 - unloading with INTERNAL_FORMAT 110
 - Unloading with records tagged with 'HEADER' and columns tagged with their name 164
 - unloading XML data 103

- examples (*continued*)
 - unloading XML format with
 - AUTOTAG 162
 - unloading XML format with SQL
 - statement and SQL expressions 163
 - unloading XML with SQL statement
 - involving SQL expressions and an INTO clause 163
 - USER format 64
 - using CONCURRENT_ACCESS 108
 - Using DB2 UNLOAD JCL with the
 - HPU load module 484
 - using LISTDEF and TEMPLATE 203
 - using REFORMAT 204
 - using SELECT statements 62
 - using TEMPLATESET 203
 - VARIABLE format 63
 - WITH UR 174
- EXCLUDE keyword 84
- EXEC statement 76
- EXECUTE keyword 110
- EXPDL keyword 89
- expressions 174
- EXTERNAL format
 - description 2
 - examples 65
- EXTERNAL keyword 190

F

- fast select block syntax 174
- fast unload syntax 463
- FIC
 - See* full image copies (FIC)
- filtering data 30
- first-time customization 31
- FlashCopy 110
- FLOAT keyword 134
- FORMAT block syntax 190
- FORMAT keyword 190
- FORMAT primary command 213
- FORMAT TYPE
 - XMLSET format 134
- format types
 - DATE 459
 - TIME 461
 - TIMESTAMP 461
- FORMAT USER 134
 - default settings 134
- formats
 - DELIMITED 2, 63
 - DSNTIAUL 2, 63
 - EXTERNAL 2, 65
 - REORG unload-only 2, 61
 - USER 2, 64
 - VARIABLE 2, 63
- FROM keyword 174
- full image copies (FIC)
 - examples 59
- FUSER primary command 213

G

- GBLPARAL parameter 447
- GDGLIMIT keyword 89

- general options
 - specifying 218
- General Options panel 218, 249
- GENERAL primary command 213
- GLOBAL OPTIONS block syntax 105

H

- hardware requirements 8
- HIDDEN parameter 134
- hiding panel text 29

I

- ICxx-ICxxxxnnn dname 79
- IFERROR parameter 134
- image copies
 - full 59
 - incremental 59
 - last full 60
 - unload consistency 71
- in predicate 174
- INCLUDE keyword 84
- incremental copies
 - examples 59
- INDEX keyword 84
- INDEXSCAN keyword 110
- INDEXSPACE keyword 84
- INDEXSPACES keyword 84
- INFEXIT 257
- INFEXIT member 257
- INFVSMIG parameter 440
- input data
 - unloading LOB 68
 - unloading XML 68
- input data sources 57
 - full image copies (FIC) 59
 - incremental image copies 59
 - last full image copies 60
 - nonpartitioned table spaces 57
 - partitioned table spaces 57
- installation library dsnames 395
- interactive component
 - displaying LOADDDN file
 - parameters 248
 - displaying OUTDDN files list 245
 - excluding partitions 221
 - generating
 - batch utility jobs 14
 - JCL 249
 - including partitions 221
 - modifying LOADDDN file
 - parameters 248
 - modifying SELECT statement USER
 - format 241
 - overview 213
 - primary commands 213
 - processing long names 233
 - saving JCL into file 250
 - selecting
 - tables 231
 - selecting columns 238
 - selecting formats
 - date 252
 - SELECT statement 234
 - time 252

- interactive component (*continued*)
 - selecting formats (*continued*)
 - timestamp 253
 - selecting mask formats 254
 - sorting SELECT statement
 - columns 240
 - specifying options
 - general 218
 - SELECT statement 225
 - specifying parameters
 - COPYDDN file 222
 - UNLOAD file 224
 - specifying SELECT statement
 - format 234
 - starting
 - DB2 Admin Tool 215
 - DB2 Tools Launchpad 217
 - stand-alone mode 215
 - updating SELECT statements list 228
 - updating the job card 251
- INTERNAL keyword 190
- INTO keyword 174
- INZ0005 panel 217
- INZ0010 panel 217
- INZA10 panel 218
 - generating JCL 249
 - saving JCL into file 250
- INZA201 panel 221
- INZA21 panel 221
- INZA30 panel 222
- INZA40 panel 224
- INZA501 panel 225
- INZA502 panel 225
- INZA601 panel 228
- INZA61 panel 231
- INZADBI member 54
- INZB101E panel 228
- INZB101I panel 234
- INZB102E panel 228
- INZB102I panel 234
- INZB12 panel 234
- INZB201 panel 238
- INZB21 panel 238
- INZB40 panel 240
- INZB501 panel 241
- INZB502 panel 241
- INZB503 panel 241
- INZB504 panel 241
- INZB51 panel 241
- INZB52 panel 241
- INZB601I panel 245
- INZB61 panel 245
- INZB70I panel 248
- INZDB21X 55
- INZDB21X member 456
- INZEXECU member 76
- INZEXIT 259
- INZLINK 395
- INZLOAD 395
- INZP006 panel
 - DATE 252
 - MASK 254
 - TIME 252
 - TIMESTAMP 253
- INZP041 panel 233
- INZP057 panel 251
- INZP060 panel 218

INZPLIB ddname 80
INZPLIB library 395
ISPF messages library 395
ISPF messages parameter 395
ISPF panels library 395
ISPF panels parameter 395
ISPF skeletons library 395
ISPF skeletons parameter 395
ISPF tables library 395
ISPF tables parameter 395

J

JCL
 running two SELECT statements
 sample 76
JCL (job control language)
 generating
 interactive component 249
JCL primary command 213
JOBLIB ddname 80
JUST parameter 134

K

keyboard shortcuts 10
keywords
 190
 @ variable 89
 &variable 89
 ASCII 168
 ASIS 168
 AUTOTAG 134
 BUFNO 89
 CCSID 168
 character expression 89
 CHECK_INTEGRITY 110
 CLONE 110
 CLONED 84
 COL 198
 COPYDDN 110
 DATABASE 84
 DATACLAS 89
 DATE 134
 DATEDELIM 134
 DB2 110
 DDLDDN 110
 DELIMITED 190, 195
 DIR 89
 DISP 89
 DSN 89
 DSNTIAUL 190, 193
 DSNTYPE 89
 EBCDIC 168
 EXCLUDE 84
 EXECUTE 110
 EXPDL 89
 EXTERNAL 190
 FLOAT 134
 FORMAT 190
 FROM 174
 GDGLIMIT 89
 INCLUDE 84
 INDEX 84
 INDEXSCAN 110
 INDEXSPACE 84

keywords (*continued*)
INDEXSPACES 84
INTERNAL 190
INTO 174
LAST_IC 110
length 89
LENGTH 134
LENGTHBYTE 134
LIST 84, 110, 174
LISTDEF 84
LISTDEFDD 83
LOADDDN 190
LOADINDDN 134
LOADOPT 134
LOCK 110
MAXPART 110
MAXPRIME 89
MGMTCLAS 89
MODELDCB 89
NBRSECND 89
NULL 134
NULLID 134
NULLPAD 134
NULLPOS 134
OBID_REPORT 110
OFF 83
ORDER BY 174
ORDER CLUSTER 174
ORIGINOBID 174
OUTDDN 168
OUTEXIT 168
OUTFREQROWS 168
OUTMAXROWS 168
PADDING 134
PARALLELISM 110
parenthetical expression 89
PART 110, 174
PARTLEVEL 84
PCTPRIME 89
PIC 134
QSAM BUFFERS 110
QUIESCE 110
QUIESCECAT 110
REFORMAT 134
RETPD 89
RI 84
SPACE 89
SPANNED 134
SQLPART 174
STACK 89
start 89
STORCLAS 89
TABLE 84
TABLESPACE 84
TABLESPACES 84
TAPERREPORT 110
TAPEUNIT 110
TEMPLATE 89
TEMPLATEDD 83
TEMPLATESSET 134
TIME 134
TIMEDELIM 134
TIMESTAMP 134
TRTCH 89
TYPE 198, 458
UNCNT 89
UNICODE 168

keywords (*continued*)
UNIT 89
UNLDDN 110
UNLFREQROWS 110
UNLMAXROWS 110
UNLROWSET 134
USER 190, 198
using with TEMPLATE
 statements 102
VARIABLE 190, 196
VOLCNT 89
VOLUMES 89
VSAM BUFFERS 110
WITH UR 174
XML 190, 199
XMLSET 134

L

last full image copies (FIC)
 example 60
LAST_IC keyword 110
Launchpad
 integrating DB2 HPU 54
 starting DB2 HPU interactive
 component 217
LBFnnnnn ddname 79
legal notices
 cookie policy 487
 notices 487
 programming interface
 information 487
 terms and conditions 489
 trademarks 487
length keyword 89
LENGTH keyword 134
LENGTH parameter 134
LENGTHBYTE keyword 134
LENGTHBYTE parameter 134
libraries
 APF LOAD modules 395
 authorizing SINZLINK 13
 DBRM 395
 dsnames 395
 INZPLIB 395
 ISPF messages 395
 ISPF panels 395
 ISPF skeletons 395
 ISPF tables 395
 LIBUTIL 395
 LOAD modules 395
 PARMLIB 395
 SINZDBRM 395
 SINZLINK 395
 SINZLOAD 395
LIBUTIL library 395
like predicate 174
LIKE table-name 193
LIMUNIT parameter 440
LIST keyword 84, 110, 174
LISTDEF
 examples 203
LISTDEF block syntax 84
LISTDEF keyword 84
LISTDEF partition processing 88
LISTDEF specification 88
LISTDEFDD keyword 83

- LOAD modules library 395
- LOAD modules parameter 395
- LOADDDN
 - statement example 76
- LOADDDN keyword 190
 - using with TEMPLATE statements 102
- LOADDDN panel 248
- LOADDDN primary command 213
- LOADINDDN keyword 134
- LOADOPT
 - parameter example 76
- LOADOPT keyword 134
- LOB columns
 - unloading 204
- LOB input data 68
- LOB output data 68
- LOCK keyword 110
- LOCK parameter 218
- logical unload
 - description 61
 - DSNTIAUL example 63
 - performing 62
 - specifying LISTDEF 88
- LOWMEM parameter 447

M

- maximizing information on panels 29
- maximizing performance
 - See* performance
- MAXPART keyword 110
- MAXPRIME keyword 89
- MAXSORT parameter 453
- MAXTUNIT parameter 440
- members
 - INFEXIT 257
 - INZADBI 54
 - INZDB21X 456
 - INZEXECU 76
- memory
 - decreasing 265
- messages
 - return codes 333
- metadata library
 - maintenance, best practices 20
 - overview 18
 - specifying 25
- MGMTCLAS keyword 89
- MODELDCB keyword 89
- modifying settings 25
- MySupport 10

N

- NBRSECND keyword 89
- nonnumeric data 459
- nonpartitioned table spaces 57
- notices 487
- NULL
 - specifying by using interactive component 225
- NULL keyword 134
- NULL parameter 134
- null predicate 174

- NULLID
 - specifying by using interactive component 225
- NULLID keyword 134
- NULLID parameter 134
- NULLPAD keyword 134
- NULLPOS
 - specifying by using interactive component 225
- NULLPOS keyword 134
- NULLPOS parameter 411
- numeric data 458

O

- OBID_REPORT keyword 110
- OFF keyword 83
- OPALLFMT parameter 411
- options 29
 - specifying 218
- OPTIONS block syntax 134
- OPTIONS primary command 213
- ORDER BY keyword 174
- ORDER CLUSTER keyword 174
- ORDERBY primary command 213
- ORIGINOBID keyword 174
- OUTDDN File panel 245
- OUTDDN keyword 168
 - using with TEMPLATE statements 102
- OUTDDN List panel 245
- OUTDDN primary command 213
- OUTEXIT keyword 168
- OUTFREQROWS keyword 168
- OUTMAXROWS keyword 168
- output data
 - consistency 69
 - DATE type 459
 - nonnumeric 459
 - numeric 458
 - TIME type 461
 - TIMESTAMP type 461
 - types 458
 - unloading LOB 68
 - unloading XML 68
- output data consistency
 - unloading 71
- output data sets 168
- output encoding scheme 65
- output files
 - allocating 66
 - calling a user exit 67
 - limiting rows unloaded 67
 - overriding 66
 - partitions 66
 - specifying one output file per partition 66
- output formats 61
 - DELIMITED 2, 195
 - DELIMITED format 63
 - DSNTIAUL 2, 193
 - DSNTIAUL format 63
 - encoding scheme 65
 - EXTERNAL 2, 65
 - logical unload 62
 - physical unload 61
 - REORG unload-only 2, 61

- output formats (*continued*)
 - specifying parameters 411
 - USER 2, 64, 198
 - VARIABLE 2, 63, 196
 - XML 199
- output types
 - TEMPLATESET specification 134
- overview 1

P

- PADDING keyword 134
- PADDING parameter 134
- panel display options 29
- panels
 - Copy DB2 Entries 50
- PARALLEL parameter 447
- parallel processing 8, 57, 80, 110, 168
- PARALLELISM keyword 110
- parameter block 257
- parameters 395
 - ACCPREP 447
 - ACTLGUSR 397
 - APF LOAD modules 395
 - BLOBF 134
 - CCSID 134
 - COLLOBJT 397
 - CTRLLIBM 411
 - customizing DB2 397
 - DATEDEL 411
 - DB2PARAL 397
 - DBCLOBF 134
 - DBRM 395
 - DELIM 134
 - DFSIGDCB 411
 - DISPLUSR 397
 - DTDELIM 411
 - DTNULDLM 411
 - ENFORCE_COLUMN_ORDER 134
 - GBLPARAL 447
 - HIDDEN 134
 - IFERROR 134
 - INFVSMIG 440
 - ISPF messages 395
 - ISPF panels 395
 - ISPF skeletons 395
 - ISPF tables 395
 - JUST 134
 - LENGTH 134
 - LENGTHBYTE 134
 - LIMUNIT 440
 - LOAD modules 395
 - LOWMEM 447
 - MAXSORT 453
 - MAXTUNIT 440
 - NULL 134
 - NULLID 134
 - NULLPOS 411
 - OPALLFMT 411
 - PADDING 134
 - PARALLEL 447
 - PARMLIB 395
 - PGDFIN 397
 - PIC block 134
 - PLANOBJT 397
 - PLANOWN 397
 - QSBUFFNO 440

parameters (continued)

QUIESCAT 397
 QUIESUSR 397
 QUITUSR 397
 RETRYMOD 447
 RETRYNB 447
 RETRYW 447
 RTSESTIM 397
 SCUNMOD 439
 SCUNSUB 439
 SCUNTSO 439
 SIZE 447
 SORTUTIL 453
 SPANNED 134
 SQLACCES 397
 SRTVNBRE 453
 SRTVSMAX 453
 SRTVSMIN 453
 STRNGCST 411
 SUBTKSOF 447
 TAPEUNIT 440
 TEMPLATE 134
 TIMEDEL 411
 TEMPLDSCB 440
 ULACCTRL 397
 ULAUTAG 411
 ULCHKCPY 397
 ULCNVTRC 397
 ULDATE 411
 ULDEGREE 397
 ULFORMAT 411
 ULHIDDEN 397
 ULIDENT 411
 ULLIKE 397
 ULLOCK 397
 ULMASK 411
 ULMAXPAR 447
 ULNULL 411
 ULOCSIZE 447
 ULOPTLDP 411
 ULOPTLDT 411
 ULOPTLX 397
 ULOPTNS 397
 ULPIC 411
 ULQSCE 397
 ULQSCEBH 397
 ULROWSET 447
 ULRTSID 397
 ULRTTST 397
 ULSEDB2 397
 ULSIGDTP 411
 ULTIME 411
 ULTMSTP 411
 ULTR32K 411
 ULTRIM 411
 UNLDDDN 411
 UNLLDER 397
 UNLSCHEM 397
 UTLPARAL 447
 VBUFSIZE 447
 VIZ003 395
 VIZ004 395
 VIZ007 395
 VIZ012 395
 VIZ013 395
 VIZ015 395
 VIZ016 395

parameters (continued)

VIZ017 395
 VIZ018 395
 VSBUFND 440
 VUA007 440
 VUM011 397
 VUM012 397
 VUM014 397
 VUM018 440
 VUM020 397
 VUM022 440
 VUM023 440
 VUM024 453
 VUM025 453
 VUM026 453
 VUM027 397
 VUM028 397
 VUM029 411
 VUM030 397
 VUM031 397
 VUM032 397
 VUU011 397
 VUU012 397
 VUU013 397
 VUU014 411
 VUU015 411
 VUU016 411
 VUU017 411
 VUU018 411
 VUU019 411
 VUU020 411
 VUU021 397
 VUU022 411
 VUU023 397
 VUU025 438
 VUU026 438
 VUU027 438
 VUU028 397
 VUU029 411
 VUU030 397
 VUU031 411
 VUU032 411
 VUU033 397
 VUU034 397
 VUU035 447
 VUU036 447
 VUU038 411
 VUU039 397
 VUU040 411
 VUU041 447
 VUU042 397
 VUU043 411
 VUU044 447
 VUU045 411
 VUU046 411
 VUU047 447
 VUU048 447
 VUU049 447
 VUU054 411
 VUU055 411
 VUU056 411
 VUU057 411
 VUU058 411
 VUU059 411
 VUU060 447
 VUU061 447
 VUU062 397

parameters (continued)

VUU063 397
 VUU064 411
 VUU065 397
 VUU066 397
 VUU067 397
 VUX003 447
 VUX004 447
 VUX005 453
 VUX006 453
 VUX007 447
 VUX008 447
 VUX009 447
 VUX010 440
 VUX011 397
 VUX016 440
 VUX017 440
 VUX019 440
 VUX022 447
 VUX023 397
 VUX025 447
 VUX030 447
 VUX031 397
 VUX032 440
 VUX033 440
 VUX036 397
 VUX037 453
 VZM005 440
 VZM006 439
 VZM007 439
 VZM008 439
 VZM010 440
 WAITQTY 447
 WAITQTYM 447
 WAITUNIT 447
 WRKMXPR 440
 WRKSPACE 453
 WRKTUNIT 440
 WRKTCVNT 440
 WRKUNTSW 440
 WRKVOL 440
 parenthetical expression keyword 89
 PARMLIB library 395
 PARMLIB parameter 395
 PART keyword 110, 174
 PART primary command 213
 Partition List panel 221
 partition processing 88
 partitions
 specifying one output file 66
 PARTLEVEL keyword 84
 PARTLEVEL partition processing 88
 PCTPRIME keyword 89
 performance
 buffering 8
 maximizing 8
 parallel processing 8
 synchronization technique 8
 PGDFIN parameter 397
 physical unload
 description 61
 specifying LISTDEF 88
 PIC block parameter 134
 PIC keyword 134
 PIC sign
 specifying by using interactive
 component 225

- PLANOBJT parameter 397
- PLANOWN parameter 397
- predicates
 - basic 174
 - between 174
 - in 174
 - like 174
 - null 174
- preparing to use Tools Customizer 25
- primary commands
 - COLUMNS 213
 - COPYDDN 213
 - FORMAT 213
 - FUSER 213
 - GENERAL 213
 - JCL 213
 - LOADDDN 213
 - OPTIONS 213
 - ORDERBY 213
 - OUTDDN 213
 - PART 213
 - SELECT 213
 - SELECTALL 213
 - UNLDDN 213
 - WHERE 213
- privileges 75
- problems
 - diagnostic information about 267
- process 3
- PROCESS OPTIONS block syntax 83
- processing complex SELECT statements 168
- Processing mode considerations 78
- programming interface information 487

Q

- QSAM BUFFERS keyword 110
- QSBUFNO parameter 440
- QUIESCAT parameter 397
- QUIESCE keyword 110
- QUIESCE parameter 218
- QUIESCECAT keyword 110
- QUIESCECAT parameter 218
- QUIESUSR parameter 397
- QUITSUSR parameter 397

R

- RACF READ authority 75
- reader comment form 5
- REFORMAT
 - example 204
- REFORMAT keyword 134
- registers
 - specifying supported 174
 - USER exit communication 257
- REORG unload-only format 61
- requirements
 - hardware 8
 - software 8
- reserved words
 - using 110
- RETPD keyword 89
- RETRYMOD parameter 447
- RETRYNB parameter 447

- RETRYW parameter 447
- RI keyword 84
- roadmaps
 - customizing for the first time 31
 - first-time customization 31
- rows
 - limiting unloaded 67
- RTSESTIM parameter 397

S

- sample parameters
 - unload 411
- SAVE JCL panel 250
- scalar functions 174
- scenarios 6
- screen readers and magnifiers 6, 10
- SCUNMOD parameter 439
- SCUNSUB parameter 439
- SCUNTSO parameter 439
- search condition 174
- SELECT block syntax 168
- SELECT Columns panel 238
- SELECT Format panel 228, 234
- SELECT Order By panel 240
- SELECT primary command 213
- SELECT statement
 - examples 76, 200, 202
- SELECT Statement List panel 228
- SELECT Statement Option panel 225
- SELECT statements 110
 - overview 168
 - processing complex 168
- Select User Format panel 241
- SELECTALL primary command 213
- service information 5
- SINZDBRM library 395
- SINZLINK library
 - authorizing 13
 - description 395
- SINZLOAD library 395
- SIZE parameter 447
- software requirements 8
- SORT ddname 79
- sorting data 30
- SORTUTIL parameter 453
- SPACE keyword 89
- SPANNED keyword 134
- SPANNED parameter 134
- special registers 174
- specifying data sets 25
- SQLACCES parameter 397
- SQLPART keyword 174
- SRTVNBRE parameter 453
- SRTVSMAX parameter 453
- SRTVSMIN parameter 453
- STACK keyword 89
- stand-alone mode
 - starting 215
- standard DB2 type data 458
- start keyword 89
- statements
 - EXEC 76, 78
 - SELECT 76, 110, 200
 - UNLOAD TABLESPACE 110
- STEPLIB ddname 80
- STORCLAS keyword 89

- STRNGCST parameter 411
- subsystems
 - specifying 217
- SUBTKSOF parameter 447
- summary of changes 1
- support
 - required information 267
- support information 5
- supported conversions 462
- SVC 99 89
- synchronization technique 8
- syntax compatibility 463
- syntax compatibility
 - Fast Unload 463
 - UNLOAD PLUS 469
- syntax diagrams
 - DELIMITED block 195
 - DSNTIAUL block 193
 - fast select block 174
 - FORMAT block 190
 - GLOBAL OPTIONS block 105
 - LISTDEF block 84
 - main 82
 - OPTIONS block 134
 - PROCESS OPTIONS block 83
 - SELECT block 168
 - TEMPLATE block 89
 - UNLOAD block 110
 - USER block 198
 - VARIABLE block 196
 - XML block 199
- SYSABEND ddname 80
- SYSIN ddname 80
- SYSPRINT
 - ddname 80
 - example 209
- SYSPRINT example 209
- SYSTEMM ddname 80

T

- TABLE keyword 84
- Table List panel 231
- table spaces
 - nonpartitioned input data 57
 - partitioned input data 57
 - selecting 217
- TABLESPACE keyword 84
- TABLESPACES keyword 84
- tape devices 110
- tape options 89
- TAPEREREPORT keyword 110
- TAPEUNIT keyword 110
- TAPEUNIT parameter 440
- technotes 5, 10
- TEMPLATE
 - examples 203
- TEMPLATE block syntax 89
- TEMPLATE keyword 89
- TEMPLATE parameter 134
- TEMPLATE statements
 - keywords 102
 - using to unload data from a table with LOBs 104
 - using to unload data from a table with XML 103
- TEMPLATEDD keyword 83

- TEMPLATESSET
 - processing multiple UNLOAD and SELECT statements 203
- TEMPLATESSET keyword 134
- terminology 8
- TIME format types
 - specifying output data 461
- TIME keyword 134
- TIMEDEL parameter 411
- TIMEDELIM keyword 134
- TIMESTAMP format types
 - specifying output data 461
- TIMESTAMP keyword 134
- TMPDSCB parameter 440
- Tools Customizer
 - associated list
 - adding DB2 entries 37
 - overview 15
 - associating DB2 entries 37
 - browsing parameters 50
 - changing display options 29
 - component 15
 - Copy DB2 Entries panel 50
 - copying DB2 entries 50
 - Create a DB2 Entry panel 37
 - creating DB2 entries 37
 - customization jobs
 - deleting 53
 - displaying 52
 - generating 46
 - maintaining 53
 - renaming 53
 - sort sequence 47
 - submitting 47, 52
 - customization library
 - deleting jobs 53
 - maintaining 53
 - recustomizing 53
 - renaming jobs 53
 - customization library qualifier
 - specifying 25
 - customization templates 455
 - Customized status 15
 - Customizer Workplace panel 46
 - customizing a new version of a product 31
 - customizing a product for the first time 31
 - customizing conversion
 - parameters 439
 - customizing DB2 HPU DB2
 - parameters 397
 - customizing file management
 - parameters 440
 - customizing output data
 - parameters 411
 - customizing sort parameters 453
 - customizing tuning parameters 447
 - data sets
 - customization library 18
 - data store 18, 20
 - DATASTOR 20
 - Discover EXEC library 18
 - metadata library 18, 20
 - data store data set
 - specifying 25

- Tools Customizer (*continued*)
 - DB2 data sharing members
 - adding 37
 - associating 37
 - copying 50
 - creating 37
 - DB2 entries 15
 - adding 37
 - associating 37
 - copying 50
 - creating 37
 - defining 46
 - deleting 51, 52
 - generating jobs for 46
 - removing 51
 - selecting 46
 - specifying 46
 - unassociating 51, 52
 - DB2 group attach field
 - specifying 25
 - DB2 group attach names
 - adding 37
 - associating 37
 - copying 50
 - creating 37
 - DB2 parameters
 - defining 44
 - editing 44
 - DB2 Parameters panel 44
 - DB2 subsystems
 - adding 37
 - associating 37
 - copying 50
 - creating 37
 - defining DB2 parameters 44
 - defining LPAR parameters 41
 - defining parameters 39, 46
 - defining product parameters 39
 - deleting DB2 entries 52
 - deleting jobs 32
 - Discover Customized Product
 - Information panel 35
 - customizing parameters 395
 - Discover EXEC
 - customizing a new version of a product 31
 - retrieving product information automatically 35
 - Discovered status 15
 - discovering product information 35
 - displaying jobs 52
 - displaying panel text 29
 - editing LPAR parameters 41
 - editing product parameters 39
 - Errors in Customization status 15
 - features 3
 - finding trace data set 267
 - Finish Product Customization
 - panel 47
 - first-time customization 31
 - generating jobs 46
 - hiding panel text 29
 - high-level qualifier 15
 - Incomplete status 15
 - integrating DB2 HPU into DB2
 - Admin 438
 - job sort order 47

- Tools Customizer (*continued*)
 - jobs
 - deleting 53
 - displaying 52
 - maintaining 53
 - renaming 53
 - submitting 52
 - LPAR Parameters panel 41
 - maintaining jobs 53
 - master list
 - adding DB2 entries 37
 - Associate DB2 Entry for Product panel 37
 - overview 15
 - maximizing information on
 - panels 29
 - metadata libraries 33
 - specifying 33
 - metadata library
 - specifying 25
 - multiple instances 25
 - multiple-LPAR environment 53
 - Not Required status 15
 - overview 3
 - panels
 - Associate DB2 Entry for Product 37
 - Copy DB2 Entries 50
 - Create a DB2 Entry 37
 - Customizer Workplace 46
 - DB2 Parameters 44
 - Discover Customized Product Information 35
 - Finish Product Customization 47
 - LPAR Parameters 41
 - Product Parameters 39
 - Specify the Metadata Library 33
 - parameters
 - browsing 50
 - viewing 50
 - preparing to use 25
 - product 15
 - product parameters
 - changing 32
 - editing 32
 - modifying 32
 - Product Parameters panel 39
 - customizing conversion
 - parameters 439
 - customizing DB2 Admin
 - parameters 438
 - customizing DB2 HPU DB2
 - parameters 397
 - customizing file management
 - parameters 440
 - customizing library dsnames 395
 - customizing output data
 - parameters 411
 - customizing sort parameters 453
 - customizing tuning
 - parameters 447
 - Ready to Customize status 15
 - recustomization 31
 - recustomizing a product 31, 32
 - removing DB2 entries 51
 - roadmaps
 - recustomizing a product 32

- Tools Customizer (*continued*)
 - Specify the Metadata Library panel 33
 - specifying metadata libraries 33
 - starting 24
 - status types
 - Customized 15
 - Discovered 15
 - Errors in Customization 15
 - Incomplete 15
 - Not Required 15
 - Ready to Customize 15
 - submitting jobs 47
 - terminology 15
 - trace data set 267
 - troubleshooting 267
 - user job card settings
 - specifying 25
 - viewing parameters 50
 - trace data set
 - finding 267
 - trademarks 487
 - translations 65
 - troubleshooting
 - decreasing memory 265
 - TRTCH keyword 89
 - TYPE keyword 198, 458
 - Type Values panel
 - DATE 252
 - MASK 254
 - TIME 252
 - TIMESTAMP 253
- ## U
- ULACCTRL parameter 397
 - ULAUTAG parameter 411
 - ULCHKCPY parameter 397
 - ULCNVTRC parameter 397
 - ULDATE parameter 411
 - ULDEGREE parameter 397
 - ULFORMAT parameter 411
 - ULHIDDEN parameter 397
 - ULIDENT parameter 411
 - ULLIKE parameter 397
 - ULLOCK parameter 397
 - ULMASK parameter 411
 - ULMAXPAR parameter 447
 - ULNULL parameter 411
 - ULOCSIZE parameter 447
 - ULOPTLDP parameter 411
 - ULOPTLDT parameter 411
 - UOPTLX parameter 397
 - UOPTNS parameter 397
 - ULPIC parameter 411
 - ULQSCCE parameter 397
 - ULQSCCEBH parameter 397
 - ULROWSET parameter 447
 - ULRTSID parameter 397
 - ULRTTST parameter 397
 - ULSEDB2 parameter 397
 - ULSIGDTP parameter 411
 - ULTIME parameter 411
 - ULTMSTP parameter 411
 - ULTR32K parameter 411
 - ULTRIM parameter 411
 - UNCNT keyword 89
 - UNICODE keyword 168
 - UNIT keyword 89
 - UNLDDDN parameter 411
 - UNLDDN
 - specifying file parameters 224
 - UNLDDN File Parameters panel 224
 - UNLDDN keyword 110
 - using with TEMPLATE statements 102
 - UNLDDN primary command 213
 - UNLFREQROWS keyword 110
 - UNLFREQROWS parameter 218
 - UNLLDER parameter 397
 - UNLMAXROWS keyword 110
 - UNLMAXROWS parameter 218
 - UNLOAD block syntax 110
 - UNLOAD command 110
 - overview 82
 - unloading nonpartitioned table spaces 57
 - unload parameters 411
 - unload plus syntax 469
 - UNLOAD TABLESPACE statement 110
 - UNLROWSET keyword 134
 - UNLSCHEM parameter 397
 - user abend codes 329
 - USER block syntax 198
 - user exit
 - calling 67
 - communication 257
 - overview 257
 - parameter block 257
 - USER format
 - creating sequential data sets 200
 - description 2
 - examples 64
 - overview 198
 - User Format Zoom panel 241
 - User Job Card panel 251
 - user job card settings
 - specifying 25
 - USER keyword 190, 198
 - utilities
 - management solutions 9
 - utility management
 - comprehensive solutions 9
 - UTLPARAL parameter 447
 - UTPRINT ddname 80
- ## V
- VARIABLE block syntax 196
 - VARIABLE format
 - creating DB2 LOAD-compatible sequential data sets 202
 - description 2
 - examples 63
 - overview 196
 - VARIABLE keyword 190, 196
 - VBUF SIZE parameter 447
 - VIZ003 parameter 395
 - VIZ004 parameter 395
 - VIZ007 parameter 395
 - VIZ012 parameter 395
 - VIZ013 parameter 395
 - VIZ015 parameter 395
 - VIZ016 parameter 395
 - VIZ017 parameter 395
 - VIZ018 parameter 395
 - VOLCNT keyword 89
 - VOLUMES keyword 89
 - VSAM BUFFERS keyword 110
 - VSBUFND parameter 440
 - VUA007 parameter 440
 - VUM011 parameter 397
 - VUM012 parameter 397
 - VUM014 parameter 397
 - VUM018 parameter 440
 - VUM020 parameter 397
 - VUM022 parameter 440
 - VUM023 parameter 440
 - VUM024 parameter 453
 - VUM025 parameter 453
 - VUM026 parameter 453
 - VUM027 parameter 397
 - VUM028 parameter 397
 - VUM029 parameter 411
 - VUM030 parameter 397
 - VUM031 parameter 397
 - VUM032 parameter 397
 - VUU011 parameter 397
 - VUU012 parameter 397
 - VUU013 parameter 397
 - VUU014 parameter 411
 - VUU015 parameter 411
 - VUU016 parameter 411
 - VUU017 parameter 411
 - VUU018 parameter 411
 - VUU019 parameter 411
 - VUU020 parameter 411
 - VUU021 parameter 397
 - VUU022 parameter 411
 - VUU023 parameter 397
 - VUU025 parameter 438
 - VUU026 parameter 438
 - VUU027 parameter 438
 - VUU028 parameter 397
 - VUU029 parameter 411
 - VUU030 parameter 397
 - VUU031 parameter 411
 - VUU032 parameter 411
 - VUU033 parameter 397
 - VUU034 parameter 397
 - VUU035 parameter 447
 - VUU036 parameter 447
 - VUU038 parameter 411
 - VUU039 parameter 397
 - VUU040 parameter 411
 - VUU041 parameter 447
 - VUU042 parameter 397
 - VUU043 parameter 411
 - VUU044 parameter 447
 - VUU045 parameter 411
 - VUU046 parameter 411
 - VUU047 parameter 447
 - VUU048 parameter 447
 - VUU049 parameter 447
 - VUU054 parameter 411
 - VUU055 parameter 411
 - VUU056 parameter 411
 - VUU057 parameter 411
 - VUU058 parameter 411
 - VUU059 parameter 411
 - VUU060 parameter 447

VUU061 parameter 447
 VUU062 parameter 397
 VUU063 parameter 397
 VUU064 parameter 411
 VUU065 parameter 397
 VUU066 parameter 397
 VUU067 parameter 397
 VUX003 parameter 447
 VUX004 parameter 447
 VUX005 parameter 453
 VUX006 parameter 453
 VUX007 parameter 447
 VUX008 parameter 447
 VUX009 parameter 447
 VUX010 parameter 440
 VUX011 parameter 397
 VUX016 parameter 440
 VUX017 parameter 440
 VUX019 parameter 440
 VUX022 parameter 447
 VUX023 parameter 397
 VUX025 parameter 447
 VUX030 parameter 447
 VUX031 parameter 397
 VUX032 parameter 440
 VUX033 parameter 440
 VUX036 parameter 397
 VUX037 parameter 453
 VZM005 parameter 440
 VZM006 parameter 439
 VZM007 parameter 439
 VZM008 parameter 439
 VZM010 parameter 440

W

WAITQTY parameter 447
 WAITQTYM parameter 447
 WAITUNIT parameter 447
 what's new 1
 WHERE clause
 search condition 174
 WHERE primary command 213
 WITH UR keyword 174
 WRKMXPR parameter 440
 WRKSPACE parameter 453
 WRKTUNIT parameter 440
 WRKTVCNT parameter 440
 WRKUNTSW parameter 440
 WRKVOL parameter 440

X

XML block syntax 199
 XML format
 overview 199
 XML input data 68
 XML keyword 190, 199
 XML output data 68
 XML parameter 134
 XMLSET keyword 134

Z

Zoom Partition panel 221



Product Number: 5655-AA1

Printed in USA

SC27-6557-02

