



CICS for MVS/ESA

SC33-1167-02

Operations and Utilities Guide

Version 4 Release 1

Note!

Before using this information and the product it supports, be sure to read the general information under "Notices" on page vii.

Second edition (April 1997)

This edition applies to Version 4 Release 1 of the IBM licensed program Customer Information Control System/Enterprise Systems Architecture (CICS/ESA), program number 5655-018, and to all subsequent versions, releases, and modifications until otherwise indicated in new editions. Consult the latest edition of the applicable IBM system bibliography for current information on this product.

This is the second edition of the Operations and Utilities Guide for CICS/ESA 4.1. It is based on the first edition, SC33-1167-00, which is now obsolete. Changes from the first edition are marked by the '+' sign to the left of the changes. The vertical lines in the left-hand margins indicate changes made between the CICS/ESA 3.3 edition and the CICS/ESA 4.1 first edition.

The CICS/ESA 3.3 edition remains applicable and current for users of CICS/ESA 3.3.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

At the back of this publication is a page entitled "Sending your comments to IBM". If you want to make comments, but the methods described are not available to you, please address them to:

IBM United Kingdom Laboratories Limited, Information Development,
Mail Point 095, Hursley Park, Winchester, Hampshire, England, SO21 2JN.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© **Copyright International Business Machines Corporation 1977, 1997. All rights reserved.**

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	vii
Trademarks and service marks	vii
Preface	ix
Book structure	xi
Determining if a publication is current	xi
Bibliography	xii
CICS/ESA 4.1 library	xii
Other CICS books	xiii
Books from related libraries	xiii
Systems Network Architecture (SNA)	xiii
DATABASE 2 (DB2)	xiii
DFSORT	xiii
Interactive system productivity facility (ISPF)	xiv
MVS/Enterprise Systems Architecture (MVS/ESA)	xv
Information Management System (IMS)	xvii
Summary of changes	xix
Changes for the second edition of the CICS/ESA 4.1 book	xix
Changes for the first edition of the CICS/ESA 4.1 book	xix
Changes for the CICS/ESA 3.3 edition	xx
Changes for the CICS/ESA 3.2.1 edition	xx

+
|

Part 1. Operating CICS regions	1
Operating procedures	1
Chapter 1. Overview of CICS operations	3
Starting up CICS	3
Controlling CICS operation	11
Shutting down CICS	13
Chapter 2. Starting up CICS regions	17
Specifying system initialization parameters before startup	17
Starting CICS as a batch job	18
Starting CICS as a started task	19
Overriding system initialization parameters during startup	21
System console messages for CICS startup	22
Chapter 3. Operating CICS in a multiregion environment	25
Enabling MRO	26
Opening interregion communication (IRC)	26
Defining MRO connections	26
Adding new MRO connections while CICS is running	26
Changing MRO connections while CICS is running	27
Closing interregion communication (IRC)	27
Chapter 4. CICS in an IMS data-sharing environment	29
An overview of IMS data-sharing	29
The levels of IMS data-sharing	31

Database recovery control (DBRC)	31
Internal resource lock manager (IRLM)	34
Coding CICS system initialization parameters for IMS data-sharing	34
Sample IMS data-sharing configurations	34
Overview of operating CICS in an IMS data-sharing environment	37
Operating procedures and flowcharts	42
Chapter 5. Running a batch region for DL/I shared databases	79
Coding an EXEC PGM=DFHDRP statement	81
XRF considerations	82
Chapter 6. Operating CICS in an XRF environment	83
Overview of the CICS extended recovery facility (XRF)	83
Functions of the CEMT and CEBT transactions for XRF	85
Starting up and synchronizing an XRF system	88
Shutting down CICS XRF regions	90
Takeover	94
What happens if the alternate CICS fails	111
Operating an XRF overseer	112
Chapter 7. Operating CICS from a console device	121
Entering commands from a console device	122
Using JCL to initiate CICS commands	122
Console device messages	123
Chapter 8. How to shut down CICS	127

Part 2. The CICS utility programs 131

Chapter 9. Journal utility programs	133
The DFHJUP utility program	133
The automatic journal archiving utility	146
Chapter 10. Statistics utility program, DFHSTUP	151
The statistics recording status	151
Restriction in the use of some output devices	152
Job to run the DFHSTUP program	153
Chapter 11. Trace utility print programs	163
The CICS trace utility program, DFHTU410	163
Using IPCS to print trace records written to GTF	169
Chapter 12. Dump utility programs	173
The CICS transaction dump utility program, DFHDU410	173
Using IPCS to format and analyze CICS dumps	179
Chapter 13. Monitoring utility programs	197
The monitoring dictionary utility program, DFHMNDUP	197
The sample monitoring data print program, DFH\$MOLS	202
Abend codes and error messages for the DFH\$MOLS program	212
Chapter 14. System definition file utility program, DFHCSDUP	215
Sharing the CSD between CICS/ESA 4.1 and earlier releases	215

Input to the DFHCSDUP program	216
Output from the DFHCSDUP program	216
Invoking DFHCSDUP as a batch program	217
Invoking the DFHCSDUP program from a user program	219
Commands for the DFHCSDUP program	221
ADD a group to a list	223
ALTER a resource definition	224
APPEND a list to another list	225
COPY a resource definition	227
DEFINE a resource definition	229
DELETE a resource definition, group, or list	231
EXTRACT resource definition data	233
INITIALIZE a new CSD with the required CICS resource definitions	235
LIST resource definitions	236
MIGRATE macro-defined resource definitions from tables to the CSD	238
REMOVE a group from a list	241
SERVICE a CSD	242
UPGRADE a CSD	243
VERIFY a CSD	244
Chapter 15. Identify macro-level programs utility program, DFHMSCAN	245
Overview	245
Using DFHMSCAN	246
Chapter 16. Sample resource manager in-doubt resolution program, DFH\$INDB	249
Overview of the DFH\$INDB utility	249
Job control statements to assemble and link-edit the DFH\$INDB utility	250
Job control statements to run the DFH\$INDB utility	251
Chapter 17. Signon table to RACF migration utility program, DFH\$NMIG	253
Migrating operator characteristics from the CICS SNT to the RACF database	253
Sample job stream to run the DFH\$NMIG program	254
Sample batch job to execute the CLIST	254
Chapter 18. Stagger end-of-day time sample utility program, DFH\$STED	255
Chapter 19. Message editing utility	257
Requirements	257
Installing the message editing utility	257
Using the message editing utility	259
Chapter 20. In-doubt window resolution utility, DFH\$IWUP	277
Overview of in-doubt windows	277
Overview of the in-doubt window resolution utility program, DFH\$IWUP	278
Job control statements to assemble and link-edit the DFH\$IWUP program	279
Job control statements to run the DFH\$IWUP program	280
Chapter 21. Local catalog storage program, DFH\$MUTL	287
Input	287
Output	287
Messages	287
Job control statements to run the DFH\$MUTL program	288

#

Appendix A. Operating procedures	291
Departmental and job titles	291
CICS online sessions log sheet	293
CICS incident report	295
Data set/database and transaction table	297
Transaction and program table	299
Error procedures	299
Appendix B. CICS messages that cannot be edited	305
Index	309

Notices

The following paragraph does not apply in any 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 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.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. 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 of the intellectual property rights of IBM may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

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 Laboratory Counsel, MP151, IBM United Kingdom Laboratories, Hursley Park, Winchester, Hampshire, England SO21 2JN. Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

IBM may have patents or pending patent applications covering subject matter 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 the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, New York 10594, U.S.A..

Trademarks and service marks

The following terms, used in this publication, are trademarks or service marks of IBM Corporation in the United States or other countries:

ACF/VTAM	AD/Cycle
BookManager	CICS
CICS/ESA	CICS/MVS
CICS/VSE	DATABASE 2
DB2	IBM
IMS/ESA	Language Environment
MVS/ESA	NetView
Processor Resource/Systems Manager	PR/SM
RACF	VSE/ESA
VTAM	

Preface

What this book is about

This book is intended to help you operate CICS regions in an MVS/Enterprise System Architecture (MVS/ESA) environment. It contains guidance about operating CICS regions in an MVS/ESA environment, particularly when using the extended recovery facility (XRF), IMS data sharing with CICS local DL/I, or multiregion operation (MRO). It also contains guidance about how to use the CICS batch utility programs.

If you need to know where programming interface information is described, or about the definitions of the different types of information in the CICS library, you should read the *CICS Family: Library Guide*.

This book does not describe the use of:

1. CICSplex System Manager/ESA, which you can use to control CICS regions in a CICSplex. CICSplex System Manager/ESA is available as a separate program offering, number 5695-081, and its use to control CICS regions in a CICSplex is described in the *CICSplex System Manager/ESA Operations* manual, SC33-0789.
2. The IBM CICS Transaction Affinities Utility MVS/ESA, which you can use to identify possible transaction affinities that may hinder your migration to a dynamic transaction routing environment. The IBM CICS Transaction Affinities Utility MVS/ESA is available as a separate program offering, number 5696-582, and is described in the *IBM CICS Transaction Affinities Utility MVS/ESA User's Guide*, SC33-1159.

Who is this book for?

This book is for system programmers responsible for controlling the operation of CICS regions, and planning the use of the supporting utility programs.

What you need to know to understand this book

We assume that you have experience of the (MVS/ESA) operating system, and that you are familiar with CICS, either from previous experience of the product or from reading the "Evaluation and Planning" category of manuals. (These are shown in the library diagram on page xii.)

We also assume that you are familiar with MVS* job control language (JCL) and cataloged procedures.

How to use this book

The parts and chapters of this book are self-contained. You should use an individual part or chapter where it contains information about the particular task you are engaged in. For example, see Part 2 if you need information about running one of the CICS utility programs.

Notes on terminology

Throughout this book, the following terms are used to indicate their associated meanings:

Application-owning region (AOR)

A CICS region that owns and manages application programs, through functions provided by a number of CICS control programs, principally the program control program.

File-owning region (FOR)

A CICS region whose primary purpose is to manage VSAM and BDAM files, and VSAM data tables, through function provided by the CICS file control program.

CICS

However, the formal abbreviation “CICS/ESA” is used when quoting the titles of other CICS publications.

MVS

The MVS/ESA operating system.

MVS/ESA SP 4.1

The MVS/ESA System Product Version 4 Release 1.

Queue-owning region (QOR)

A CICS region whose primary purpose is to manage CICS temporary storage queues and transient data queues, through function provided by the temporary storage control program and the transient data control program.

RACF

The MVS resource access control facility (RACF) or any other external security manager that provides equivalent function.

Resource-owning region (ROR)

A CICS region that owns more than one type of resource, such as a combined file-owning and queue-owning region.

Terminal-owning region (TOR)

A CICS region that owns and manages sessions with terminals that logon directly to the region via VTAM, using function provided by the CICS terminal control program.

XCF PR/SM policy

The function that enables MVS images to take over resources of other MVS images in the same sysplex. This term is also known as the PR/SM automatic reconfiguration facility (ARF).

\$ (the dollar symbol)

In the programming examples in this book, the dollar symbol (\$) is used as a national currency symbol. In countries where the dollar is not the national currency, the local currency symbol should be used.

Book structure

Part 1, Operating CICS regions ... page 1 to page 129

Describes how to start up, run, and shut down CICS regions.

Part 2, The CICS utility programs ... page 131 to page 285

Describes how to run the CICS utility programs, and supplies sample jobs for each utility.

Appendix A, “Operating procedures” on page 291

Describes sample operating procedures that you can adapt for your CICS environment.

Appendix B, “CICS messages that cannot be edited” on page 305

Lists those messages that you cannot change by using the message editing utility.

Determining if a publication is current

IBM regularly updates its publications with new and changed information. When first published, both hardcopy and BookManager softcopy versions of a publication are in step, but subsequent updates will probably be available in softcopy before they are available in hardcopy.

For CICS books, these softcopy updates appear regularly on the *Transaction Processing and Data Collection Kit* CD-ROM, SK2T-0730-xx. Each reissue of the collection kit is indicated by an updated order number suffix (the -xx part). For example, collection kit SK2T-0730-06 is more up-to-date than SK2T-0730-05. The collection kit is also clearly dated on the cover.

Here's how to determine if you are looking at the most current copy of a publication:

- A publication with a higher suffix number is more recent than one with a lower suffix number. For example, the publication with order number SC33-0667-02 is more recent than the publication with order number SC33-0667-01. (Note that suffix numbers are updated as a product moves from release to release, as well as for hardcopy updates within a given release.)
- When the softcopy version of a publication is updated for a new collection kit the order number it shares with the hardcopy version does not change. Also, the date in the edition notice remains that of the original publication. To compare softcopy with hardcopy, and softcopy with softcopy (on two editions of the collection kit, for example), check the last two characters of the publication's filename. The higher the number, the more recent the publication. For example, DFHPF104 is more recent than DFHPF103. Next to the publication titles in the CD-ROM booklet and the readme files, asterisks indicate publications that are new or changed.
- Updates to the softcopy are clearly marked by revision codes (usually a “#” character) to the left of the changes.

Bibliography

CICS/ESA 4.1 library

Evaluation and planning		
<i>Release Guide</i>	GC33-1161	April 1997
<i>Migration Guide</i>	GC33-1162	April 1997
General		
<i>CICS Family: Library Guide</i>	GC33-1226	April 1995
<i>Master Index</i>	SC33-1187	October 1994
<i>User's Handbook</i>	SX33-1188	April 1997
<i>Glossary (softcopy only)</i>	GC33-1189	n/a
Administration		
<i>Installation Guide</i>	GC33-1163	April 1997
<i>System Definition Guide</i>	SC33-1164	April 1997
<i>Customization Guide</i>	SC33-1165	April 1997
<i>Resource Definition Guide</i>	SC33-1166	April 1997
<i>Operations and Utilities Guide</i>	SC33-1167	April 1997
<i>CICS-Supplied Transactions</i>	SC33-1168	April 1997
Programming		
<i>Application Programming Guide</i>	SC33-1169	October 1994
<i>Application Programming Reference</i>	SC33-1170	April 1997
<i>System Programming Reference</i>	SC33-1171	April 1997
<i>Sample Applications Guide</i>	SC33-1173	October 1994
<i>Distributed Transaction Programming Guide</i>	SC33-1174	October 1994
<i>Front End Programming Interface User's Guide</i>	SC33-1175	October 1994
Diagnosis		
<i>Problem Determination Guide</i>	SC33-1176	October 1994
<i>Messages and Codes</i>	GC33-1177	April 1997
<i>Diagnosis Handbook</i>	LX33-6093	October 1994
<i>Diagnosis Reference</i>	LY33-6082	April 1997
<i>Data Areas</i>	LY33-6083	April 1997
<i>Supplementary Data Areas</i>	LY33-6081	October 1994
<i>Closely-Connected Program Interface</i>	LY33-6084	November 1996
Communication		
<i>Intercommunication Guide</i>	SC33-1181	April 1997
<i>Server Support for CICS Clients</i>	SC33-1591	February 1996
<i>CICS Family: Inter-product Communication</i>	SC33-0824	October 1996
<i>CICS Family: Communicating from CICS on System/390</i>	SC33-1697	October 1996
Special topics		
<i>Recovery and Restart Guide</i>	SC33-1182	October 1994
<i>Performance Guide</i>	SC33-1183	October 1994
<i>CICS-IMS Database Control Guide</i>	SC33-1184	October 1994
<i>CICS-RACF Security Guide</i>	SC33-1185	October 1994
<i>Shared Data Tables Guide</i>	SC33-1186	October 1994
<i>External CICS Interface</i>	SC33-1390	April 1997
<i>CICS ONC RPC Feature for MVS/ESA Guide</i>	SC33-1119	February 1996
<i>CICS Web Interface Guide</i>	SC33-1892	November 1996

The book that you are reading was republished in hardcopy format in April 1997 to incorporate updated information previously available only in softcopy. The right-hand column in the above table indicates the latest hardcopy editions of the CICS/ESA books available in April 1997. A book with a date earlier than April 1997 remains the current edition for CICS/ESA 4.1. Note that it is possible that other books in the library will be updated after April 1997.

When a new order is placed for the CICS/ESA 4.1 product, the books shipped with that order will be the latest hardcopy editions.

The style of IBM covers changes periodically. Books in this library have more than one style of cover.

For information about the softcopy books, see “Determining if a publication is current” on page xi. The softcopy books are regularly updated to include the latest information.

Other CICS books

- *CICS Application Migration Aid Guide*, SC33-0768
- *CICS Application Programming Primer (VS COBOL II)*, SC33-0674
- *CICS/ESA Facilities and Planning Guide* for CICS/ESA Version 3 Release 3, SC33-0654
- *CICS/ESA XRF Guide* for CICS/ESA Version 3 Release 3, SC33-0661
- *CICS Family: API Structure*, SC33-1007
- *CICS Family: General Information*, GC33-0155
- *IBM CICS Transaction Affinities Utility MVS/ESA*, SC33-1159

CICS Clients

- *CICS Clients: Administration*, SC33-1436
- *CICS Family: Client/Server Programming*, SC33-1435

Books from related libraries

Systems Network Architecture (SNA)

Systems Network Architecture: Function Description of Logical Unit Types, GC20-1868

Systems Network Architecture: Types of Logical Unit to Logical Unit Sessions, GC20-1869.

DATABASE 2 (DB2)

System Planning and Administration Guide, SC26-4085

Installation, SC26-4084

Sample Application Guide, SC26-4086

Guide to Publications, GC26-4111.

DFSORT

DFSORT General Information manual, GC33-4033

DFSORT Messages and Codes manual, SC26-4525.

Interactive system productivity facility (ISPF)

ISPF Dialog Management Guide, SC34-4112

ISPF Dialog Management Services and Examples, SC34-4113

ISPF/PDF Edit and Edit Macros, SC34-4121

MVS/Enterprise Systems Architecture (MVS/ESA)

Where necessary, this book references information in MVS/ESA books, using shortened versions of the book title. The following table shows the complete titles and order numbers for those MVS/ESA books:

Short title used in this book	Version 3 book	Version 4 book
<i>An Introduction to MVS</i>	<i>An Introduction to MVS</i> , GC28-1823	<i>An Introduction to MVS</i> , GC28-1823
<i>Authorized Assembler Programming Reference</i>	<i>MVS/ESA System Programming Library: Application Development Macro Reference</i> , GC28-1857	<i>MVS/ESA Application Development Reference: Services for Authorized Assembler Language Programs Volume 1</i> , GC28-1647
<i>Authorized Assembler Programming Reference Volume 2</i>	—	<i>MVS/ESA Application Development Reference: Services for Authorized Assembler Language Programs Volume 2</i> , GC28-1648
<i>Authorized Assembler Programming Reference Volume 3</i>	—	<i>MVS/ESA Application Development Reference: Services for Authorized Assembler Language Programs Volume 3</i> , GC28-1649
<i>Authorized Assembler Programming Reference Volume 4</i>	—	<i>MVS/ESA Application Development Reference: Services for Authorized Assembler Language Programs Volume 4</i> , GC28-1650
<i>General Information</i>	<i>MVS/ESA General Information for MVS/SP Version 3</i> , GC28-1359	<i>MVS/ESA General Information for MVS/SP Version 4</i> , GC28-1600
—	—	<i>MVS/ESA Implementation Guide</i> , GG24-3628
<i>Initialization and Tuning Guide</i>	<i>MVS/ESA System Programming Library: Initialization and Tuning</i> , GC28-1828	<i>MVS/ESA Initialization and Tuning Guide</i> , GC28-1634
<i>Initialization and Tuning Reference</i>	—	<i>MVS/ESA Initialization and Tuning Reference</i> , GC28-1635
<i>IPCS Command Reference</i>	<i>MVS/ESA Interactive Problem Control System (IPCS) Command Reference</i> , GC28-1834	<i>MVS/ESA Interactive Problem Control System (IPCS) Command Reference</i> , GC28-1632
<i>IPCS Customization</i>	<i>MVS/ESA Interactive Problem Control System (IPCS) Planning and Customization</i> , GC28-1832	<i>MVS/ESA Interactive Problem Control System (IPCS) Customization</i> , GC28-1630
<i>IPCS Reference Summary</i>	<i>MVS/ESA Interactive Problem Control System (IPCS) Reference Summary</i> , GC28-1835	<i>MVS/ESA Interactive Problem Control System (IPCS) Reference Summary</i> , GX22-0016
<i>IPCS User's Guide</i>	<i>MVS/ESA Interactive Problem Control System (IPCS) User's Guide</i> , GC28-1833	<i>MVS/ESA Interactive Problem Control System (IPCS) User's Guide</i> , GC28-1631
<i>JCL Reference</i>	<i>MVS/ESA JCL Reference</i> , GC28-1829	<i>MVS/ESA JCL Reference</i> , GC28-1654

Short title used in this book	Version 3 book	Version 4 book
<i>JCL User's Guide</i>	<i>MVS/ESA JCL User's Guide</i> , GC28-1830	<i>MVS/ESA JCL User's Guide</i> , GC28-1653
<i>Library Guide</i>	<i>MVS/ESA Library Guide for MVS/SP Version 3</i> , GC28-1563	<i>MVS/ESA Library Guide for MVS/SP Version 4</i> , GC28-1601
<i>Master Index</i>	<i>MVS/ESA Master Index for MVS/SP Version 3</i> , GC28-1827	<i>MVS/ESA Master Index for MVS/SP Version 4</i> , GC28-1602
<i>Planning: Dynamic I/O Configuration</i>	—	<i>MVS/ESA Planning: Dynamic I/O Configuration</i> , GC28-1674
<i>Planning: Global Resource Serialization</i>	<i>MVS/ESA Planning: Global Resource Serialization</i> , GC28-1818	<i>MVS/ESA Planning: Global Resource Serialization</i> , GC28-1621
<i>Planning: Operations</i>	—	<i>MVS/ESA Planning: Operations</i> , GC28-1625
<i>Planning: Problem Determination and Recovery</i>	<i>MVS/ESA Planning: Dump and Trace Services</i> , GC28-1838	<i>MVS/ESA Planning: Problem Determination and Recovery</i> , GC28-1629
<i>Planning: Sysplex Management</i>	—	<i>MVS/ESA Planning: Sysplex Management manual</i> , GC28-1620
<i>Problem Determination Guide</i>	<i>MVS/ESA Basics of Problem Determination</i> , GC28-1839	<i>MVS/ESA Problem Determination Guide</i> , GC28-1667
<i>Service Aids</i>	<i>MVS/ESA Service Aids</i> , GC28-1844	<i>MVS/ESA Service Aids</i> , GC28-1669
<i>System Commands</i>	<i>MVS/ESA Operations: System Commands</i> , GC28-1826	<i>MVS/ESA System Commands</i> , GC28-1626
<i>System Commands Reference Summary</i>	<i>MVS/ESA Operations: System Commands Reference Summary</i> , GX22-0013	<i>MVS/ESA System Commands Reference Summary</i> , GX22-0015
<i>System Management Facilities</i>	<i>MVS/ESA System Programming Library: System Management Facilities (SMF)</i> , GC28-1819	<i>MVS/ESA System Management Facilities (SMF)</i> , GC28-1628
<i>System Messages Volume 1</i>	<i>MVS/ESA Message Library: System Messages Volume 1</i> , GC28-1812	<i>MVS/ESA System Messages Volume 1</i> , GC28-1656
<i>System Messages Volume 2</i>	<i>MVS/ESA Message Library: System Messages Volume 2</i> , GC28-1813	<i>MVS/ESA System Messages Volume 2</i> , GC28-1657
<i>System Messages Volume 3</i>	—	<i>MVS/ESA System Messages Volume 3</i> , GC28-1658

Information Management System (IMS)

Title	IMS	
	ESA 3.1	ESA 4.1
<i>Application Programming: Design Guide</i>	SC26-4279	SC26-4279
<i>Application Programming: DC Calls</i>	SC26-4283	SC26-4283
<i>Application Programming: DL/I Calls</i>	SC26-4274	SC26-4274
<i>Application Programming: EXEC DLI Commands</i>	SC26-4280	SC26-4280
<i>Application Programming: DL/I Calls Summary</i>	SX26-3765	SX26-3765
<i>Application Programming: EXEC DLI Commands Summary</i>	SX26-3775	SX26-3775
<i>Customization Guide: Database</i>	SC26-4624	SC26-4624
<i>Customization Guide: Data Communications</i>	SC26-4625	SC26-4625
<i>Customization Guide: Systems</i>	SC26-4285	SC26-4285
<i>Database Administration Guide</i>	SC26-4281	SC26-4281
<i>Data Communication Administration Guide</i>	SC26-4286	SC26-4286
<i>Diagnosis Guide and Reference</i>	LY27-9539	LY27-9539
<i>Failure Analysis Structure Tables (FAST) for Dump Analysis</i>	LY27-9512	LY27-9512
<i>General Information</i>	GC26-4275	GC26-4275
<i>Installation Guide</i>	SC26-4276	SC26-4276
<i>Licensed Program Specifications</i>	GC26-4289	GC26-4289
<i>LU6.1 Adapter for LU6.2 Applications: Program Description/Operations</i>	SC26-4392	SC26-4392
<i>Master Index and Glossary</i>	SC26-4291	SC26-4291
<i>Messages and Codes</i>	SC26-4290	SC26-4290
<i>Operator's Reference</i>	SC26-4288	SC26-4288
<i>Operations Guide</i>	SC26-4287	SC26-4287
<i>Release Planning Guide</i>	GC26-4386	GC26-4386
<i>Sample Operating Procedures</i>	SC26-4277	SC26-4277
<i>Summary of Operator Commands</i>	SX26-3764	SX26-3764
<i>System Administration Guide</i>	SC26-4282	SC26-4282
<i>System Definition Reference</i>	SC26-4278	SC26-4278
<i>Utilities Reference</i>	SC26-4284	----
<i>Utilities Reference: Data Communication</i>	----	SC26-4628
<i>Utilities Reference: Systems</i>	----	SC26-4629

Summary of changes

+ Changes for the second edition of the CICS/ESA 4.1 book

+ This book is the second edition of the *Operations and Utilities Guide* for CICS/ESA
+ 4.1. Changes that were made for the first edition are still indicated by vertical bars
+ to the left of the changes. Changes made for this second edition are indicated by
+ the '+' symbol to the left of the changes. Users of the first edition can therefore
+ see what has changed since that first edition was published. Softcopy versions of
+ this book use both these revision indicators and use the '#' symbol to show further
+ changes since this second hardcopy edition of the book was published.

| Changes for the first edition of the CICS/ESA 4.1 book

| The major changes to CICS for CICS/ESA 4.1 that affect the first edition of this
| manual are:

| **Message editing utility program**

| A new utility program to edit CICS messages

| **DFH\$IWUP in-doubt resolution utility program**

| A new utility program to help resolve resources that have been changed by
| tasks considered to have been in-doubt after a CICS region failure.

| **IPCS SDUMP formatting program**

| New component keywords (DD, PG, TRS, and US) and parameter for
| uppercase output

| **DFHCSDUP system definition file utility program**

| New parameter for uppercase output and compatibility with definitions from
| before CICS/ESA 4.1

| **DFHDU410 transaction dump utility program**

| Renamed from DFHDUP (as named in earlier releases) and parameter for
| uppercase output

| **DFHMNDUP monitoring dictionary utility program**

| Five new control parameters (JOBNAME, JOBDATE, JOBTIME, UPPERCASE,
| and USERID)

| **DFHSTUP statistics utility program**

| Three new control parameters (SELECT TYPE, IGNORE TYPE, and
| UPPERCASE (for uppercase output)) and a new DD statement (DFHSTWRK)
| for use as a work file

| **DFHTR410 GTF trace utility program**

| Renamed from DFHTRGTF (as named in earlier releases)

| **DFHTU410 trace utility program**

| Renamed from DFHTUP (as named in earlier releases), a parameter for
| uppercase output, and enable input of multiple parameter lines.

| Also, the following changes have been made to the structure of this book:

- Chapter 1, "Overview of CICS operations" on page 3 has been added to provide an overview of operating CICS

- Chapter 2, “Starting up CICS regions” on page 17 has been added to describe how to start up CICS
- Chapter 8, “How to shut down CICS” on page 127 has been added to describe how to shutdown CICS
- Appendix A, “Operating procedures” on page 291 has been created from Chapter 1 of the CICS/ESA 3.3 edition, to move the reference information to an appendix of this book.

Changes for the CICS/ESA 3.3 edition

The major changes to CICS for CICS/ESA 3.3 that affected this manual are:

- Provision of an IPCS exit control table member, DFHIPCSP, for CICS in the new CICS410.SDFHPARM library.
- Provision of the IPCS exit routine, DFHPD330, for CICS in the new SYS1.CICS410.SDFHLINK library.
- Provision of the generic copy function for the DFHCSDUP utility.

Changes for the CICS/ESA 3.2.1 edition

The major changes to CICS for CICS/ESA 3.2.1 that affected this manual are:

- Removal of CICS internal security; CICS now depends for security on an external security manager, such as RACF
- Support for CICS operations from TSO
- Support for IMS/ESA 4.1 in CICS local DL/I
- Backup and restore of VSAM files while open for update
- Support for extended recovery in a multi-MVS environment
- Message enhancements and automation
- Changes to the CICS startup messages
- Change all the CICS library names known to SMP/E (see Table 1 on page xxi) to conform to the MVS product packaging rules, including changes to supplied JCL procedures containing references to the libraries.

<i>Table 1. CICS/ESA 4.1 library name changes</i>		
Old library name	New distribution library	New target library
CEELIB	-	SDFHC370
COBLIB	-	SDFHCOB
DCEELIB	ADFHC370	-
DCOBLIB	ADFHCOB	-
DLOADLIB	ADFHMOD	-
DPL1LIB	ADFHPL1	-
JCLLIB	ADFHINST	SDFHINST
LOADLIB	-	SDFHLOAD
LOADLIB1	-	SDFHAUTH
MACLIB	ADFHMAC	SDFHMAC
PL1LIB	-	SDFHPL1
PROCLIB	ADFHPROC	SDFHPROC
SAMPLIB	ADFHSAMP	SDFHSAMP
SOURCE	ADFHSRC	SDFHSRC
-	ADFHAPD1	SDFHAPD1
-	ADFHAPD2	SDFHAPD2
-	ADFHMSGGS	SDFHMSGGS

Under the new installation process, most CICS libraries now have both an SMP/E distribution library (ADFHxxxx) and an SMP/E target library (SDFHxxxx).

- Provision of a new CICS startup procedure, DFHSTART, that can be used to start the CICS pregenerated system, CICS MRO regions, or as a basis for your own startup procedures.
- Provision of a new CICS utility program, DFHNMIG, to transfer operator characteristics from the CICS signon table (SNT) to the RACF database.
- Provision of a new sample CICS utility program, DFH\$STED, to stagger the end-of-day time for each CICS region, to improve performance at statistics collection intervals.

Part 1. Operating CICS regions

This Part

is about operating CICS regions. It begins with an introductory chapter that provides an overview of operating CICS, and continues with more chapters about specific aspects of operating CICS.

This part contains the following chapters about operating CICS:

- Chapter 1, “Overview of CICS operations” on page 3
- Chapter 2, “Starting up CICS regions” on page 17
- Chapter 3, “Operating CICS in a multiregion environment” on page 25
- Chapter 4, “CICS in an IMS data-sharing environment” on page 29
- Chapter 5, “Running a batch region for DL/I shared databases” on page 79
- Chapter 6, “Operating CICS in an XRF environment” on page 83
- Chapter 7, “Operating CICS from a console device” on page 121
- Chapter 8, “How to shut down CICS” on page 127.

The other books in the CICS library that you may want to refer to for information related to operating CICS regions are:

- The *CICS/ESA System Definition Guide*, for information about CICS system definitions; including CICS startup JCL and system initialization parameters
- The *CICS/ESA Recovery and Restart Guide*, for information about CICS recovery and restart
- The *CICS/ESA CICS-Supplied Transactions* manual, for information about the master terminal transactions provided by CICS
- The *CICS/ESA CICS-IMS Database Control Guide*, for information about operating CICS with IMS/ESA database control (DBCTL)
- The *CICS/ESA 3.3 XRF Guide*, for general information about the CICS extended recovery facility.

If you are operating CICS within a CICSplex controlled by CICSplex SM, you should refer to the *CICSplex System Manager/ESA Operations* manual, SC33-0789.

Operating procedures

When operating CICS, you should have clearly defined operating procedures for your CICS environment. These procedures should provide information about how CICS should be operated in your CICS environment, and record actions taken while operating CICS. Some example procedures and forms for use when operating CICS are provided in Appendix A, “Operating procedures” on page 291. These are for guidance only, and you should modify the procedures and forms to suit your CICS environment.

Chapter 1. Overview of CICS operations

This chapter

provides an overview of CICS operations. It outlines:

- Starting up CICS
- Controlling CICS operations
- Shutting down CICS.

Details about CICS operations are given in Chapter 2 through Chapter 8.

Starting up CICS

When you start up CICS, you start a process called **CICS system initialization**. This process must finish before you run any transactions.

CICS system initialization involves many activities, some of which are:

- Obtaining the required storage for CICS execution from the private area in the CICS address space, above and below the 16MB line
- Setting up CICS system parameters for the run, as specified by the system initialization parameters
- Loading and initializing the CICS domains according to the start option specified by the `START=` system initialization parameter
- Loading the CICS nucleus with the required CICS modules
- Installing CICS resource definitions by:
 - Loading, from the CSD, the groups of resources specified by the `GRPLIST=` system initialization parameter
 - Loading the control tables specified by system initialization parameters.
- If `XRF=YES` is specified, signing on to the CICS availability manager (CAVM) to check that it is possible to continue initialization and perform the role requested, that is, as an active or alternate CICS region
- Opening the data sets necessary for initialization, including any needed for backout if the previous run of your CICS region was not shut down normally (except for `START=STANDBY`, when most data sets are not opened until after takeover)
- Opening BSAM sequential devices as required in the terminal control table (TCT).

If you are operating CICS with CICS recovery options, backout procedures may be used to restore recoverable resources to a logically consistent state. Backout occurs if you start CICS in one of the following ways:

- With `START=AUTO` and CICS detects that the previous shutdown was immediate or uncontrolled
- With `START=STANDBY` and `XRF=YES`, and a takeover occurs.

For background information about backout, and recovery and restart, see the *CICS/ESA Recovery and Restart Guide*.

The types of CICS startup

CICS start up can be any of the following types:

Startup

type

Effect

Cold

CICS initializes with limited reference to any system activity recorded in the CICS catalogs from a previous run of CICS. For more information, see “CICS actions on a cold start” on page 5.

Note: You can perform a fully cold start, without use of information in the CICS catalogs from any previous execution, only when the catalogs have been newly initialized. Generally, you initialize the catalogs only when you are starting your CICS region for the first time.

Warm

CICS initializes, after a normal shutdown, restoring CICS to the status it was in at the last normal CICS shutdown, except for some facilities that it initializes as for a cold start. CICS always restores the trace domain according to the system initialization parameters, and can restore other facilities depending on the COLD option of their associated system initialization parameters. For more information, see “CICS actions on a warm start” on page 5.

Emergency

CICS initializes, after an abnormal shutdown, restoring recoverable resources to their committed states. For more information, see “CICS actions on an emergency restart” on page 8.

When CICS is started up, the type of startup (and therefore the actions it takes) depends primarily on the following:

- The value of the system initialization parameter START
- Information in the CICS catalogs from previous runs of CICS.

The values of other system initialization parameters also influence the actions taken on CICS startup.

The effect of the START system initialization parameter is outlined on page 18.

For information about the types of startup, the roles of the CICS catalogs, and the effect of the START system initialization parameter, see the *CICS/ESA System Definition Guide*.

Note: You cannot explicitly request a warm or emergency restart. When selecting the type of start (using the CICS system initialization parameter, START), the choices are COLD or AUTO. AUTO can result in either a warm or an emergency restart; CICS itself determines which to use depending on the time the startup takes place.

CICS actions on a cold start

In a cold start, initialization of CICS occurs with limited reference to any system activity recorded in the CICS catalogs. No system log or warm keypoint information is used from any previous run of CICS. Dump table entries from a previous run are also deleted in a cold start.

In a cold start:

- If journaling is supported, volume profiles are built and updated.
- For journals **without** the automatic archiving option, the status of journaling is restored from information recorded in the catalog. For journals with the PAUSE option, journal status is preserved. For journals with the NOTREADY status, PAUSE messages are reissued.
- For journals **with** the automatic archiving option, the journal status, including the sequence number, is retrieved from the JACD (journal archive control data set). For those journal data sets that have the NOTREADY status, an archive job is submitted.
- TERMINAL definitions are purged from the recovery file and from the catalog.
- Existing TYPETERM and MODEL definitions are purged from the catalog.
- PROGRAM definitions are purged from the recovery file and from the catalog.
- TRANSACTION and PROFILE definitions are purged from the global catalog.
- DL/I backout records are purged from the recovery file Database status information is obtained from the DDIR table entry.
- File control records are purged from the catalog.
- Resource definition information is obtained as follows:
 - Tables specified by system initialization parameters, such as DCT=xx, are obtained from the program library.
 - Information in the groups in the list named by the GRPLIST system initialization parameter for *this* initialization is taken from the CICS system definition (CSD) file and merged with information from the program library.
 - Information in groups that have been defined or added to group lists is taken from the CSD.

CICS actions on a warm start

A warm start restores certain elements of the CICS components that can be warm started to the status that was recorded in the warm keypoint of the previous normal shutdown.

A partial warm start is similar to a complete warm start, except that some selected CICS facilities are cold-started, as specified in the system initialization parameters. Information is obtained for those facilities from the warm keypoint only if they are not specified to be cold started.

In a warm start:

- Tape volume descriptor lists are obtained from the global catalog.
- Resource definition information is obtained as follows:
 - Tables specified by system initialization parameters, such as DCT=xx, are obtained from the program library. Information contained in the warm keypoint of the previous run is used to update the information from the program library.
 - Information in the groups in the list named by the GRPLIST system initialization parameter for *this* initialization is ignored.
 - Information in the groups in the list named by the GRPLIST system initialization parameter for the *previous* initialization is obtained from the warm keypoint and the global catalog.
 - Information in groups that have been installed since the last cold start is obtained from the warm keypoint and the global catalog.
 - Information in groups that have been defined or added to group lists is taken from the CSD.
 - Information about any autoinstalled terminal that has an automatic-initiate descriptor (AID) outstanding is retrieved from the global catalog.
- Selected fields from the CSA are restored from the warm keypoint, including:
 - Region exit time interval value
 - Runaway time interval value
 - Maximum number of tasks
 - Sizes of the storage cushions for each of the dynamic storage areas:
 - CDSA
 - UDSA
 - ECDSA
 - EUDSA
 - ERDSA.
 - Extended storage cushion size
 - High-water mark number of the unit of recovery descriptor.
- The following intrapartition transient data information is restored:
 - All data held on intrapartition destinations
 - The following DCT information for each destination, from the warm keypoint:
 - Record count in the destination
 - GET and PUT pointer values
 - Trigger level values.
 - AIDs for tasks
 - The intrapartition transient data bit map.
- The following FCT information is restored to what it was at the time of the warm shutdown, using information from the global catalog:
 - The ENABLED/DISABLED/UNENABLED status
 - The SERVREQ options (UPDATE, DELETE and so on)
 - Any alterations made to the DSNAME.

- Files defined as initially OPEN are opened irrespective of their other attributes. If the file state *recovered* during initialization is ENABLED or UNENABLED, the file becomes OPEN, ENABLED after the OPEN. If the file state recovered is DISABLED, the file becomes OPEN, DISABLED.
- For local DL/I, the following data set status information is restored, if there is a corresponding DDIR entry:
 - STARTED
 - STOPPED
 - RECOVERDB
 - DUMPDB
 - ACCESS value (RO, RD, UP, or EX).

If there is no DDIR entry, the status information for that data set is discarded.

- Installed transaction and profile definitions are obtained from:
 - The groups specified in the GRPLIST system initialization parameter at the last cold start
 - The groups that have been installed since the last cold or emergency start.

The following attributes of the installed transactions and profiles are restored from the warm keypoint:

- ENABLED/DISABLED status
- Transaction priority.
- Installed program and mapset definitions are obtained from these sources:
 - The groups specified in the GRPLIST system initialization parameters at the last cold start
 - The groups that have been installed **since** the last cold start or emergency restart
 - The changes (such as LPA-eligibility) made by CEMT or EXEC CICS SET PROGRAM commands in the last run.

The ENABLED/DISABLED status of each installed program and mapset is restored from the warm keypoint. Directory information is obtained for each program and mapset during CICS initialization.

- The following TCT information is restored from the warm keypoint information:
 - Processing status (transaction, transceive, input, or receive)
 - Service status (INSERVICE or OUTSERVICE)
 - Extended attributes supported (color, programmed symbols, and so on)
 - Partition support
 - Magnetic-stripe-reader support
 - Outboard formatting support
 - Coded graphic character set identifiers
 - APL/TEXT keyboard.

If any outstanding work was scheduled for an autoinstalled terminal at the last warm shutdown, the terminal entry is recovered. (Terminal entries for autoinstalled terminals with no work outstanding are deleted at shutdown.)

- The following auxiliary temporary storage information is restored from the warm keypoint:
 - All data in the auxiliary temporary storage queues
 - The temporary storage use map.

- Interval control elements (ICEs) for outstanding START TRANSID commands are restored from the warm keypoint.
- The BMS logical messages that were created by the functions listed below but have not yet been viewed by the terminal operator are restored:
 - Message switching transaction (CMMSG).
 - ROUTE command.
 - SEND MAP ACCUM and SEND TEXT ACCUM commands, except for those messages terminated by SEND PAGE without specifying RELEASE or RETAIN. In those cases, the message might already have been viewed by the operator, but can be viewed again following the warm start.
- All unit of recovery descriptors (APPC log name, APPC resynchronization, and external resource manager) are restored from the warm keypoint, together with any associated deferred work elements (DWEs).
- The STORECLOCK value is restored from the warm keypoint.
- The intervals at which statistics were collected and status and the logical end-of-day time are restored from the global catalog.
- The monitoring status, class status and monitoring control table suffix are restored from the global catalog.
- For journals with the automatic archiving option, the journal status, including the sequence number, is retrieved from the JACD (journal archive control data set), by means of a call to the journal archive program (DFHJAP) from DFHJCRP. For those journal data sets that have the NOTREADY status, an archive job is submitted.
- If you are using standard-labeled tape journaling, and you supply the SERIES=PURGE startup parameter, CICS deletes any journal series information that is stored in the global catalog. In the absence of the SERIES startup parameter, CICS reconstructs all the journal series from the information saved in the global catalog.

If you are using disk journaling, the status of the disk journals is also saved in the global catalog. This information is used by CICS at startup to determine which journal data set is to be opened. You can use the JSTATUS=RESET system initialization parameter to cause the status in the global catalog to be ignored. During CICS startup, the status of all journal data sets is set to “ready for use”.
- Transaction and system dump table options are held in the global catalog and reapplied at a warm start.

CICS actions on an emergency restart

A CICS system that operates on resources, such as files, that have been defined by the installation to be *recoverable*, records changes to those resources in the CICS system log. If the CICS system fails, the system log at the time of failure should typically contain records of changes made by tasks that have not completed ('in-flight' tasks) and by others that have completed.

Following an abnormal termination, emergency restart collects all of the log records pertaining to in-flight tasks, and uses the information in the log records to return recoverable resources to their **committed** states—that is:

- Changes to recoverable resources made by logical units of work (LUWs) that were interrupted are backed out.
- Messages associated with message-protected tasks are preserved.
- Dynamically added resource definitions that were committed during execution of the CEDA INSTALL command are preserved.
- CICS takes an activity keypoint, so that there is at least one activity keypoint on the new system log data set. It shows that there are no in-flight tasks, and delimits the backward scan of the system log on a subsequent emergency restart, in case no other activity keypoint is written during this CICS run. If there are any journals on standard-labeled tapes, the activity keypoint includes the tape-volume descriptors. When the activity keypoint has been taken, emergency restart processing is complete.

CICS-VTAM actions after an emergency restart

When LU-LU sessions are re-established after an emergency restart (and subsequent processing), CICS participates in a resynchronization protocol with logical units to discover if any messages, in either direction, were lost when CICS was terminated.

The logical units for which resynchronization is required will have been marked in the TCTTEs. Resynchronization is *not* attempted in the following cases:

- If the terminal was acquired by a master terminal operation specifying COLDACQ.
- If the terminal was acquired with the EXEC CICS SET TERMINAL ACQSTATUS(COLDACQ) command.
- If the session is a pipeline session.
- If the TCTTE is marked to cold start the session by the TCT assembly process. This is done for terminals such as 3270 terminals that do not support the set and test sequence number (STSN) command.

Note: If the previous session abended, the use of COLDACQ overrides CICS integrity control. This could lead to data integrity problems. Also, you should check the CSMT log for an activity keypoint after the restart of a session following a CICS failure. If there is no activity keypoint, you should issue COLDACQ again after the next emergency restart.

For each logical unit that *does* require resynchronization, CICS issues an STSN command that notifies the logical unit of the sequence numbers known to CICS—that is, those numbers that backout processing placed in the TCTTE. The logical unit can compare these sequence numbers with those that it has logged for itself, and can thus determine if any messages were lost.

- If an *input* message was lost, the logical unit should retransmit it to CICS.
- If an *output* message was lost, CICS retransmits the message from the resend slot and, in so doing, deletes the resend slot.

Note: The message remains in the resend slot if CICS does *not* retransmit it. This occurs if the resynchronization process shows that the output message was not lost, or if the logical unit does not support the STSN command; the 3270 is in this category.

CICS startup and the VTAM session

In a VTAM network, the session between CICS and VTAM is started automatically if VTAM is started before CICS. If VTAM is not active when you start CICS, you receive the following messages:

```
F vtamname,USERVAR,ID=generic-applid,VALUE=specific-applid
+DFHSI1589D 'applid' VTAM is not currently active.
+DFHSI1572 'applid' Unable to OPEN VTAM ACB - RC=xxxxxxx, ACB CODE=yy.
```

Although the MODIFY NET, USERVAR command is only significant when you are running CICS with XRF, the USERVAR message occurs for both XRF=YES and XRF=NO CICS systems. If you receive messages DFHSI1589D and DFHSI1572, and if the CICS region is not initializing as an alternate CICS region, you can start the CICS-VTAM session manually when VTAM is eventually started, by means of the CEMT SET VTAM OPEN command from a supported MVS console or a non-VTAM terminal.

If VTAM is active, but CICS still cannot open the VTAM ACB because VTAM does not recognize the CICS APPLID, you receive the following messages:

```
F vtamname,USERVAR,ID=generic-applid,VALUE=specific-applid
+DFHSI1592I 'applid' CICS applid not (yet) active to VTAM.
+DFHSI1572 'applid' Unable to OPEN VTAM ACB - RC=00000008, ACB CODE=5A.
```

This may be caused by an error in the value of APPLID operand, in which case you must correct the error and restart CICS. For information about other causes and actions, see the *CICS/ESA Messages and Codes* manual.

Concurrent initialization of VTAM and XRF alternate CICS regions

An XRF alternate CICS region cannot initialize properly until it has successfully opened the VTAM ACB.

Because VTAM and the alternate CICS region may be initialized concurrently, it is possible that several tries may have to be made to open the VTAM ACB. If VTAM is not active, the following message is written to the system console every 15 seconds:

```
DFHSI1589D 'applid' VTAM is not currently active.
```

If VTAM is active, but CICS cannot open the VTAM ACB, the following messages are written to the system console:

```
+DFHSI1572 'applid' Unable to OPEN VTAM ACB - RC=xxxxxxx, ACB CODE=yy.
DFHSI1590 'applid' XRF alternate cannot proceed without VTAM.
```

CICS abends with a dump (abend code 1590).

End of CICS startup

Whichever type of startup is performed, when the message:

```
DFHSI1517 - 'applid': Control is being given to CICS.
```

is displayed on the operating system console, CICS is ready to process terminal requests. (*applid* is the value of the specific APPLID system initialization parameter.)

When the startup process is completed, users are able to enter transactions from any terminals that are connected to CICS. For information about the CICS-supplied transactions, see the *CICS/ESA CICS-Supplied Transactions* manual.

Controlling CICS operation

While CICS is running, you can control its operation by changing CICS system definitions and by deleting and installing resource definitions.

Note: You cannot change CICS system definition values set by *some* system initialization parameters during CICS startup. To change such values, you must specify the new values on system initialization parameters, and restart CICS with those changed system initialization parameters.

CICS supplies a number of transactions that you can use to control CICS and its resources while it is running. It also supplies a variety of utility programs, some of which you can use to help with system management.

Controlling CICS with CICSplex SM

If you are running your CICS regions in a CICSplex, you can use CICSplex System Manager/ESA functions to control the operation of CICS; that is to:

- Change CICS system attributes
- Reset CICS' date and time to match those of the operating system
- Rebuild security profiles for CICS
- Write CICS statistics to an SMF data set
- Add, remove, or reset CICS system and transaction dump codes.

For information, see the *CICSplex System Manager/ESA Operations* manual, SC33-0789.

CICS-supplied transactions

CICS supplies a number of transactions that you can use to control CICS and its resources while it is running. CICS-supplied transactions have identification codes that start with the letter C and are four characters long.

The most significant transactions for CICS operation are CEMT, CEST, and CEDA. The following sections outline these three transactions. For information about these and other CICS transactions, see the *CICS/ESA CICS-Supplied Transactions* manual.

CEMT

CEMT is the master terminal transaction. You can use the CEMT transaction to view the values of CICS system definitions and to change such definitions while CICS is running. You can also use CEMT to manage databases, in particular for the dynamic allocation and deallocation of data sets.

With CEMT, you can:

- Control the number of tasks, or the number of certain types of task, running at any given time
- Purge tasks from the system
- Enable or disable transactions
- Enable or disable files; for example, to allow controlled access to it by application programs
- Start or stop tracing (you can also use CETR for this), monitoring, or statistics activities
- Switch dump data sets when one is full
- Open and close interregion communication connections
- Install newly link-edited copies of application programs
- Specify some messages (usually urgent ones) to be routed to the master terminal.

To view the values of CICS system definitions, use the CEMT INQUIRE command.

To change the values of system definitions, or to change CICS operation, use the CEMT SET, CEMT PERFORM, or CEMT DISCARD command.

Note: CEMT is a powerful tool and its use can significantly affect your system, and its users. Therefore, you should give the transaction adequate security protection in a production CICS region.

CEST

CEST is the supervisor terminal transaction. It provides a subset of the CEMT function. The CEST INQUIRE and SET commands enable you to inquire about and alter some of the system values of control units, lines, netnames, tasks and terminals.

CEDA

You can use the CEDA transaction to:

- View resource definitions
- Change existing resource definitions
- Create new resource definitions

on the CSD that your CICS region is using. You can also use the CEDA transaction to install resource definitions into a running CICS region.

Similarly, you can use the CEDB transaction to view, change, or create resource definitions, and can use the CEDC transaction to view resource definitions.

For information about the CEDA, CEDB, and CEDC transactions, see the *CICS/ESA Resource Definition Guide*.

CICS-supplied utility programs

CICS supplies a number of utility programs to help you manage your system. These utility programs are described in Part 2, “The CICS utility programs” on page 131.

Shutting down CICS

This section describes the three types of CICS system shutdown (normal, immediate, or uncontrolled) and the events that cause them. For information about CICS shutdown, see the *CICS/ESA Recovery and Restart Guide*.

- In normal shutdown, CICS performs a controlled sequence of operations that leave the system in a well-defined state. Existing tasks are allowed to finish.

The following events can cause normal shutdown of CICS:

- Using the CEMT PERFORM SHUTDOWN transaction
- Using the EXEC CICS PERFORM SHUTDOWN command.

- In immediate shutdown, CICS remains in overall control, but it does a minimum of processing so the system can be terminated rapidly. Existing tasks are not allowed to finish, and could abend.

The following events can cause immediate shutdown of CICS:

- Using the CEMT PERFORM SHUTDOWN IMMEDIATE transaction
- Using the EXEC CICS PERFORM SHUTDOWN IMMEDIATE command
- A CICS system abend
- A program check.

- In uncontrolled shutdown, CICS is not given the chance to do any processing once the event causing it to terminate has occurred.

The following events can cause uncontrolled shutdown of CICS:

- Power failure
- Machine check
- Operating system failure.

After a normal shutdown, it is possible to warm start CICS. After an immediate or an uncontrolled shutdown, an emergency restart or a cold start must be performed.

CICS XRF systems

When an XRF active CICS region is terminated abnormally, the alternate CICS region normally completes initialization and takes over. However, the alternate CICS region also terminates if the user has initiated a normal shutdown of the active CICS region and has *not* specified that takeover is to take place.

Normal shutdown (PERFORM SHUTDOWN)

Normal shutdown is initiated by the master terminal operator or by an application program, and is accomplished in three stages. (In comparison, immediate shutdown is accomplished by termination processing.)

First stage of normal shutdown

During the first stage of CICS normal shutdown, all terminals are active and all CICS facilities are available. The following actions take place concurrently:

- Message DFHTM1715 is issued to the console and the master terminal user to inform the operator that CICS is terminating.
- The local and global catalogs are closed.
- Tasks that already exist will complete. (Long running tasks, such as conversational tasks, must end before system can complete.)
- Tasks to be automatically initiated will run, if they can start before the second stage.
- Any user-written programs listed in the first part of the shutdown program list table (PLT) are run sequentially.
- The Front End Programming Interface (FEPI) is requested to shut down.
- The terminal that initiated the shutdown, if any, is detached. This allows the operator to start any further tasks that might be required, or to purge any tasks.

A new task started as a result of terminal input is allowed to start only if it has been defined as SHUTDOWN(ENABLED) in its TRANSACTION resource definition or if the transaction identifier is listed in the current transaction list table (XLT). The XLT list of transactions restricts the tasks that can be started by terminals and allows the system to shut down in a controlled manner. The current XLT is the one specified by the XLT=xx system initialization parameter, which may be overridden by the XLT option of the CEMT or EXEC CICS PERFORM SHUTDOWN command.

Certain CICS-supplied transactions are, however, allowed to start whether or not their code is listed in the XLT. These transactions are CEMT, CESF, CLS1, CLS2, CSAC, CSTE, and CSNE.

Note: You should not change the SHUTDOWN(ENABLED) attribute of the resource definitions for these transactions, otherwise CICS may not shut down successfully.

- A request is issued to all interregion communication (IRC) activity.
- Terminal control is requested to ignore all further input.
- If this is a non-XRF system, CLSDST requests are issued for all VTAM terminals.
- The termination task waits for all terminal activity to cease, before entering the second stage of shutdown.

The first shutdown stage is complete when the last of the programs specified in the first part of the shutdown PLT has run and all user tasks are complete.

Second stage of normal shutdown

During the second stage of shutdown, terminals are not active, and no new tasks are allowed to start. The following processing takes place:

1. User-written programs listed in the second part of the shutdown PLT (if any) are executed sequentially. These programs cannot communicate with terminals, or make any request that would cause a new task to start.
2. If DL/I is active, DL/I processing is started.

3. All currently open CICS files are now closed.
4. The AD/Cycle Language Environment/370 partition is shut down.
5. Journal control is shutdown.
6. The transient data CI buffer and the temporary storage buffers are flushed.
7. CICS writes the following information to the global catalog:
 - A warm keypoint. This contains information that is used to restore the operating environment during a subsequent warm start.
 - The tape volume descriptor lists for all journals (if any) that are implemented on standard labeled tapes—including the system log.
 - A warm-start-possible indicator. This status applies on the next initialization of CICS if START=AUTO is specified.

The global catalog domain cannot complete its processing until this information has been recorded, as the global catalog must remain open until it is done.
8. Transient data is terminated.
9. A dump is taken, if one is required.
10. If TAKEOVER was specified on the command to shut down an XRF CICS region, a “signoff abnormal” request is made from the CICS availability manager (CAVM).
11. The following message is issued:


```
DFHKE1799 applid TERMINATION OF CICS/ESA IS COMPLETE
```
12. CICS completes some internal processing, then returns control to MVS.

Immediate shutdown (PERFORM SHUTDOWN IMMEDIATE)

During immediate shutdown of CICS, possibly requested by the master terminal operator or an application program, processing is different from a normal shutdown in the following important ways:

- User tasks are not guaranteed to complete.
- None of the programs listed in the shutdown PLT are run.
- CICS does *not* write a warm keypoint or a warm-start-possible indicator to the global catalog.
- CICS does not close files defined to CICS file control.

To preserve data integrity, the next initialization of CICS **must** be an emergency restart. If the next initialization of CICS specifies START=AUTO, there will be an emergency restart.

The processing involved in immediate shutdown is described as CICS system termination processing. (In comparison, normal shutdown involves processing.)

Unlike processing, controls are not exercised to ensure that resources and services remain available as long as they are needed. One consequence of this is that transaction and CICS system abends can occur during immediate shutdown. Thus, if a task tries to use a resource that has already been terminated, the task abends. Then dynamic transaction backout is invoked, and that might also fail because it could also try to use a resource that has been terminated.

In addition, if CICS system termination processing is delayed significantly, tasks in the system waiting for input from terminals that are no longer available are likely to extend beyond the period for deadlock timeout specified in the DTIMOUT option of the TRANSACTION definition.

First stage of immediate shutdown

During the first stage of an immediate shutdown, the following processes take place:

1. The system termination task drives the collection of termination statistics.
2. If there is a terminal associated with the event that caused the immediate shutdown, a message is sent to inform the operator that CICS is terminating.
3. If the shutdown request has arrived by transaction routing, the associated terminal is freed.
4. Terminal input is no longer accepted.
5. The Front End Programming Interface (FEPI) is requested to shut down immediately.

Second stage of immediate shutdown

During the second stage of an immediate shutdown, the following processing takes place:

1. DL/I is terminated.
2. Journal control is shut down.
3. Transient data is terminated.
4. A dump is taken, if requested.
5. Interregion sessions are terminated.
6. If CICS is signed on to the CICS availability manager (CAVM), a “signoff abnormal” request is made from CAVM.
7. The local catalog and global catalog are left to be closed by the operating system.
8. The following message is issued:
DFHKE1799 *applid* TERMINATION OF CICS/ESA IS COMPLETE
9. CICS completes some internal processing, then returns control to MVS.

Uncontrolled shutdown

An uncontrolled shutdown of CICS can be caused by a power failure, a machine check, or an operating system failure.

In each case, CICS cannot perform any shutdown processing. In particular, CICS does *not* write a warm keypoint or a warm-start-possible indicator to the global catalog.

To preserve data integrity, the next initialization of CICS **must** be an emergency restart. If the next initialization of CICS specifies START=AUTO, there will be an emergency restart.

Chapter 2. Starting up CICS regions

This chapter

describes *how* to start up CICS regions. It assumes that any customization of CICS, the generation of any additional support required, and all the necessary CICS system definitions have already been carried out.

For an overview of CICS startup, see “Starting up CICS” on page 3.

For information about defining CICS systems, see the *CICS/ESA System Definition Guide*. For example, that book describes the system initialization parameters in detail, and describes how to create a CICS startup job stream.

You can start CICS in either of two ways:

- Use the MVS START command to start CICS as a started task. (See “Starting CICS as a started task” on page 19.)
- Submit a CICS batch job to the MVS internal reader. (See “Starting CICS as a batch job” on page 18.)

Whichever method you use to start CICS, you determine how CICS starts up, and the facilities and resources that it can use, by specifying values for system initialization parameters to be used by the CICS startup procedure. You would normally specify the system initialization parameters that CICS is to use before you start CICS. (See “Specifying system initialization parameters before startup.”) However, once you have started the initialization of CICS, you can override the system initialization parameters specified before startup; for example, to enable a specific facility for that run of CICS. (See “Overriding system initialization parameters during startup” on page 21.)

Specifying system initialization parameters before startup

You would normally specify the system initialization parameters that CICS is to use in the following ways, before starting CICS:

1. In the system initialization table, loaded from a library in the STEPLIB concatenation of the CICS startup procedure
2. In the PARM parameter of the EXEC PGM=DFHSSIP statement of the CICS startup procedure
3. In the SYSIN data set defined in the startup procedure (but only if SYSIN is coded in the PARM parameter)

The system initialization parameters are processed in the preceding order, with later system initialization parameter values overriding those specified earlier.

In particular, you can specify a new value for the START system initialization parameter, which can have any of the following values:

START=AUTO

If you specify the START=AUTO system initialization parameter, CICS determines, by inspecting the control record in the global catalog, whether to perform a cold, warm, or emergency start.

START=AUTO should be the normal mode of operation, with the choice of start being made by CICS automatically.

START=COLD

If you specify the START=COLD system initialization parameter, CICS performs a cold start, but with some reference to the previous run of a CICS system using the same global catalog.

START=STANDBY

CICS starts up as an XRF alternate CICS region, by beginning an emergency restart then suspending until it needs to perform a takeover.

Note: You must also specify the XRF=YES system initialization parameter.

START=LOGTERM

CICS restarts only to put an end of file label on the system log, and then terminates *before* doing any backout processing.

Note: This option is intended for use only when you are running CICS with XRF=NO.

For example, if your CICS startup procedure specifies:

```
//INITCICS EXEC PGM=DFHSIP,REGION=&REG,  
//          PARM=('CLONE=HT##,SYSIDNT=HTH1,SIT=6$,DCT=SP,SYSIN,CN')  
//*  
//SYSIN DD DISP=SHR,DSN=CICS410.CICSH###.SYSIN(CICS&CLONE)
```

CICS uses system initialization parameters from the following sources, with later system initialization parameters overriding earlier ones:

1. The system initialization table, DFHSIT6\$, from the STEPLIB concatenation
2. The DCT=SP system initialization parameter from the PARM parameter
3. The member CICSH### of the CICS410.CICSH###.SYSIN data set
4. The system console.

In the above example, if you specify DCT=2\$ in the SYSIN data set, and then enter DCT=3\$ at the system console, CICS loads DFHDCT3\$.

Starting CICS as a batch job

To start CICS as a batch job, you submit the job through the internal reader. Your CICS startup job can contain the CICS startup procedure inline, or can invoke a cataloged startup procedure. This latter method has the advantage that several CICS startup jobs (for example, for different CICS regions) can use the same procedure, tailoring the procedure through startup parameters. For example, Figure 1 on page 19 shows a CICS startup job that invokes the cataloged procedure, CICSTASK, to cold start a terminal-owning with the startup parameters SYSIDNT=HTH1 and CLONE=HT##1. By altering the SYSIDNT and CLONE

parameters, the same job could be used to start other CICS regions with the same procedure.

```
//CIDCTOR JOB (accounting information),userid,MSGCLASS=A,MSGLEVEL=(1,1),
//          CLASS=C,NOTIFY=userid
//*****
//* THIS JOB CAN BE USED TO START UP A CICS REGION
//*****
//*
//CICS410 EXEC CICSTASK,
// START=COLD,
// SYSIDNT='HTH1',          SYSID OF CICS REGION
// CLONE='HT##'            CLONE CICS REGION TYPE
//*
```

Figure 1. Job to start a CICS TOR, HTH1

In this example of the MVS START command:

- CICSTASK is the name of a cataloged CICS startup procedure, tailored from the CICS-supplied sample startup procedure.
- SYSIDNT is the qualifier used to identify CICS system data sets that are unique to each CICS region.
- CLONE is the qualifier of the member in the SYSIN data set, CICS410.SYSIN, that has system initialization parameters unique to each *type* of CICS region.

For information about the CICS-supplied startup procedure, see the *CICS/ESA Installation Guide*.

Starting CICS as a started task

To start CICS as a started task, you use the MVS START command. For example, to start CICS from the MVS console:

```
S[START procname[.identifier] [,SUB=subsystemname] [,keyword=option
                                     [,keyword=option] . . .]
```

procname

The name of the cataloged procedure that defines the CICS job to be started.

identifier

The name you choose to identify the CICS task.

SUB=subsystemname

The name of the subsystem that is to select the job for processing. If you omit this parameter, the primary job entry subsystem is used.

keyword=option

Any appropriate keyword to override the corresponding parameter in the procedure. You can use this parameter to override symbolic parameters defined in the cataloged procedure.

Notes:

1. Using this method, your startup job stream must be coded according to the rules for coding procedures, and the procedure must be installed in an MVS procedure library.
2. You must do either of the following:
 - Give the MVS started task procedure a name different from the subsystem name in IEFSSNaa (default 'CICS')
 - Issue the START command with the parameter SUB=JES2 or SUB=JES3 as appropriate.

For information about the complete syntax of the START command, and all the keywords and options you can use, see the *MVS/ESA System Commands* manual.

To start CICS, you only need to code **procname.identifier,keyword(s)=option**.

For example, you could use the following start command to start the CICS tasks listed in Figure 2:

```
START CICS410
```

```
//CICS410 PROC
//*
//DUMMY EXEC PGM=IEFBR14
//*
// START CICS410.CICSHTH1,SYSIDNT='HTH1',CLONE='HT##'
//*      START=COLD
// START CICS410.CICSHAH1,SYSIDNT='HAH1',CLONE='HA##'
//*      START=COLD
// START CICS410.CICSHAH2,SYSIDNT='HAH2',CLONE='HA##'
//*      START=COLD
// START CICS410.CICSHRH1,SYSIDNT='HRH1',CLONE='HR##'
//*      START=COLD
//*
//* END OF CICS START PROCEDURE
```

Figure 2. Procedure to start a CICS TOR, two AORs, and an ROR

In this example of the MVS START command:

- CICSTASK is the name of a cataloged CICS startup procedure, tailored from the CICS-supplied sample startup procedure.
- The following CICS regions are started:
 - Terminal-owning region, CICSHTH1
 - Application-owning region, CICSHAH1
 - Application-owning region, CICSHAH2
 - Resource-owning region, CICSHRH2
- SYSIDNT is the qualifier used to identify CICS system data sets that are unique to each CICS region.
- CLONE is the qualifier of the member in the SYSIN data set, CICS410.SYSIN, that has system initialization parameters unique to each *type of* CICS region.

For information about the CICS-supplied startup procedure, see the *CICS/ESA Installation Guide*.

If you are running CICS with RACF, you must associate the cataloged procedure name with a suitably authorized RACF user through the RACF table, ICHRIN03. For details about this association, see the *CICS/ESA CICS-RACF Security Guide*.

Overriding system initialization parameters during startup

Once you have started the initialization of CICS, you may want to override system initialization parameters specified in the SIT, PARM parameter, and SYSIN data set of the CICS startup procedure. You can do this by specifying new values for system initialization parameters at the system console.

Note: You can specify system initialization parameters at the system console only if the CONSOLE keyword was specified in either the PARM parameter or in the SYSIN data set.

If you specify the CONSOLE (or CN) keyword in the PARM statement of the EXEC PGM=DFHSIP statement or SYSIN data set of your CICS startup procedure, CICS prompts you to enter system initialization parameters at the system console.

Generally, CICS does not begin to read from the console until it has loaded the SIT and processed any initialization parameters that are coded in the PARM parameter and the SYSIN data set. CICS accepts system initialization parameters from the console until you terminate the input with .END.

Through the console, you can specify a SIT system initialization parameter only as the first parameter when prompted by message DFHPA1921, at which point CICS tries to load the specified SIT. If you try to specify a SIT system initialization parameter after CICS has loaded the SIT it is rejected as an error.

Rules for coding parameters at the console

When it is ready to read parameters from the console, CICS displays the following message (where nn is the reply ID):

```
nn DFHPA1104 applid - SPECIFY ALTERNATIVE SIT PARAMETERS, IF ANY,  
AND THEN TYPE '.END'.
```

You can enter as many initialization parameters as you can get on one line of the console, but you must use a comma to separate parameters. CICS continues to prompt for system initialization parameters with displays of message DFHPA1105 until you terminate console input by entering the .END control keyword.

Entering corrections to parameters at the console

If you have coded PARMERR=INTERACT, and CICS detects a parameter error, either in the keyword or in the value you have assigned to it, CICS prompts you to correct the error with message DFHPA1912 or DFHPA1915:

```
DFHPA1912 'applid' SIT OVERRIDE 'keyword' IS NOT RECOGNIZED.  
SPECIFY CORRECT SIT OVERRIDE.  
DFHPA1915 'applid' INVALID DATA HAS BEEN DETECTED FOR SIT OVERRIDE 'keyword'.  
RESPECIFY THE OVERRIDE.
```

CICS prompts you to enter corrections to any errors it finds in the PARM parameter or the SYSIN data set **after** it has loaded the SIT, and as each error is detected. This means that if there is an APPLID parameter **following** the parameter that is in error, either in the PARM parameter or in the SYSIN data set, it is the APPLID coded in the SIT that CICS displays in messages DFHPA1912 and DFHPA1915.

System console messages for CICS startup

This section gives some typical message sequences that are displayed when starting the terminal-owning region, CICSHTH1, using the procedure shown in Figure 2 on page 20. (Messages for the other regions in the CICSplex are not given, because they are effectively identical.)

For messages from the startup and takeover of XRF CICS regions, see the *CICS/ESA Installation Guide*.

The CICS-supplied default system initialization table, DFHSIT, was used (the SIT system initialization parameter was not specified) and system initialization parameters specific to the CICSHTH1 region were supplied from a permanent SYSIN data set; you can see these options in Figure 3, because the SYSIN data is displayed on the console.

These systems included COBOL2 and PL/I support by default.

Notes are given after Figure 3 to annotate specific messages of interest.

```
JES2 JOB LOG -- SYSTEM MVSH -- NODE WINMVSH
JOB04877
↓
14.55.13 . IRR010I USERID userid IS ASSIGNED TO THIS JOB.
14.55.13 . ICH70001I userid LAST ACCESS AT 14:55:02 ON FRIDAY, APRIL 15, 1994
14.55.13 . $HASP373 C IDCITH1 STARTED - INIT 2 - CLASS C - SYS MVSH
14.55.13 . IEF403I IDCITH1 - STARTED - TIME=14.55.13
14.55.14 . 1 DFHPA1102 DBDCCICS READING OVERRIDE PARAMETERS FROM SYSIN.
14.55.14 . DFHPA1927 DBDCCICS * *****
14.55.14 . DFHPA1927 DBDCCICS * CICS system initialization parameters common to all clone TORs *
14.55.14 . DFHPA1927 DBDCCICS * *****
14.55.14 . 2 DFHPA1927 DBDCCICS APPLID=CICSHTH1 The CICS application identifier
14.55.14 . 3 DFHPA1927 DBDCCICS CICS SVC=218 The default CICS SVC number
14.55.14 . 4 DFHPA1927 DBDCCICS CWAKEY=CICS CICS key for the CWA
14.55.14 . DFHPA1927 DBDCCICS DBP=1$ Dynamic backout program - no local DL/I support
14.55.14 . DFHPA1927 DBDCCICS DCT=2$ The default/sample destination control table
14.55.14 . 5 DFHPA1927 DBDCCICS DFLTUSER=IVPUSER Default userid is IVPUSER
14.55.14 . 6 DFHPA1927 DBDCCICS DTRPGM=WMDYP Dynamic transaction routing required
14.55.14 . DFHPA1927 DBDCCICS FCT=NO No file control table (using RDO for files)
14.55.14 . 7 DFHPA1927 DBDCCICS GMTEXT='You are on CICS/ESA 4.1 Terminal-Owning Region (TOR) - CICSHTH1'
14.55.14 . 8 DFHPA1927 DBDCCICS GRPLIST=(CICSHT#1,IPLLIST) Initialize with TOR and common group lists
14.55.14 . DFHPA1927 DBDCCICS * The IRC & ISC parameters required for MRO
14.55.14 . 9 DFHPA1927 DBDCCICS IRCSTRT=YES Start IRC during initialization
14.55.14 . DFHPA1927 DBDCCICS ISC=YES Include the intersystem communication program
14.55.14 . DFHPA1927 DBDCCICS JCT=2$ The CICS sample journal control table
14.55.14 . DFHPA1927 DBDCCICS MXT=32 Set maximum tasks to 32
14.55.14 . 10 DFHPA1927 DBDCCICS PGAIPGM=ACTIVE Program autoinstall is active
14.55.14 . 11 DFHPA1927 DBDCCICS RENTPGM=PROTECT Read-only ERDSA required.
14.55.14 . DFHPA1927 DBDCCICS SEC=NO Switch off security
14.55.14 . DFHPA1927 DBDCCICS SPOOL=YES SPOOL interface open for STAT
14.55.14 . DFHPA1927 DBDCCICS SRT=1$ The CICS sample System Recovery Table
14.55.14 . 12 DFHPA1927 DBDCCICS STGPROT=YES Storage protection required
14.55.14 . DFHPA1927 DBDCCICS TCT=5$ Sequential terminals required
14.55.14 . 13 DFHPA1927 DBDCCICS TCTUKEY=CICS CICS key for TCT User Areas
14.55.14 . DFHPA1927 DBDCCICS TRTABSZ=200 Trace table size
```

Figure 3 (Part 1 of 2). Console messages for startup of TOR – CICSHTH1

```

14.55.14 . DFHPA1927 DBDCCICS XRF=NO                Start a non-XRF active CICS region
14.55.14 . DFHPA1103 DBDCCICS END OF FILE ON SYSIN.
14.55.14 . DFHPA1101 DBDCCICS DFHSIT  IS BEING LOADED.
14.55.14 . DFHPA1108 DBDCCICS DFHSIT  HAS BEEN LOADED. (GENERATED AT: MM/DD= 02/09 HH:MM= 21:49).
14.55.14 . +DFHTR0103 TRACE TABLE SIZE IS 200K
14.55.14 . 14 +DFHSM0122I CICSHTH1 Limit of DSA storage below 16MB is 5,120K.
14.55.14 . +DFHSM0123I CICSHTH1 Limit of DSA storage above 16MB is 20M.
14.55.14 . 12 +DFHSM0115I CICSHTH1 Storage protection is active.
14.55.14 . 15 +DFHSM0126I CICSHTH1 Transaction isolation is not active.
14.55.14 . +DFHDM0101I CICSHTH1 CICS is initializing.
14.55.15 . +DFHSI1500 CICSHTH1 CICS/ESA Version 4.1.0 Startup is in progress.
14.55.15 . +DFHXS1100I CICSHTH1 Security initialization has started.
14.55.15 . +DFHSI1501I CICSHTH1 Loading CICS nucleus.
14.55.16 . +DFHDU0304I CICSHTH1 Transaction Dump Data set DFHDMPA opened.
14.55.16 . +DFHXS1102I CICSHTH1 Security is inactive.
14.55.16 . +DFHXS1101I CICSHTH1 Security initialization has ended.
14.55.18 . 16 +DFHMN0105I CICSHTH1 Using default Monitoring Control Table.
14.55.18 . +DFHMN0110I CICSHTH1 CICS Monitoring is inactive.
14.55.18 . +DFHSI1502I CICSHTH1 CICS startup is Cold.
14.55.18 . +DFHSI1503I CICSHTH1 Terminal data sets are being opened.
14.55.18 . +DFHSI1510I CICSHTH1 Journal control subtask is being attached/entered.
14.55.19 . +DFHCP0101I CICSHTH1 CPI initialization has started.
14.55.19 . +DFHPR0104I CICSHTH1 Partner resource manager initialization has started.
14.55.20 . +DFHFC0100I CICSHTH1 File Control initialization has started.
14.55.20 . +DFHTD0100I CICSHTH1 Transient Data initialization has started.
14.55.20 . +DFHSI1516I CICSHTH1 Opening journal data sets.
14.55.20 . +DFHAI0101I CICSHTH1 AITM initialization has started.
14.55.20 . +DFHJC4508 CICSHTH1 CICS SYSTEM LOG. PRIMARY  DATA SET NOW RECEIVING OUTPUT ON 68C
14.55.20 . +DFHJC4500 CICSHTH1 01 of 01 journals successfully opened
14.55.20 . +DFHFC0101I CICSHTH1 File Control initialization has ended.
14.55.20 . +DFHTD0101I CICSHTH1 Transient Data initialization has ended.
14.55.20 . +DFHCP0102I CICSHTH1 CPI initialization has ended.
14.55.20 . +DFHPR0105I CICSHTH1 Partner resource manager initialization has ended.
14.55.20 . +DFHAI0102I CICSHTH1 AITM initialization has ended.
14.55.20 . 8 +DFHSI1511I CICSHTH1 Installing group list CICSIT#1.
14.55.23 . +DFHSI1511I CICSHTH1 Installing group list IPLLIST .
14.55.25 . +DFHSI1519I CICSHTH1 The interregion communication session was successfully started.
14.55.25 . +DFHAP1204I CICSHTH1 COBOL2 is being initialized.
14.55.25 . +DFHAP1205I CICSHTH1 C/370 is being initialized.
14.55.25 . +DFHSI1517 CICSHTH1 Control is being given to CICS.

```

Figure 3 (Part 2 of 2). Console messages for startup of TOR – CICSHTH1

Notes:

- 1 The APPLID in messages DFHPA1102I and DFHPA1101I is shown as DBDCCICS, even though there is an APPLID parameter in SYSIN which specifies a different name. However, the SYSIN parameters have not yet been processed; CICS has only scanned the SYSIN data set for a SIT= parameter, displaying the SYSIN parameters on the console as it does so. The individual SYSIN parameters are processed only after the SIT has been loaded. Later messages which include the APPLID show the name correctly as CICSHTH1.
- 2 This is the SYSIN parameter that specifies the applid CICSHTH1.
- 3 The SVC value of 218 is only an example; you should specify the SVC value to be used, defined in your SVC table.
- 4 CICS obtains storage for the CWA in CICS key. (CICS is operating with storage protection, specified by STGPROT=YES.) This means that only programs executing in CICS key can modify the CWA, and user-key programs have read-only access.

5 You must define the default CICS userid, by the SIT parameter DFLTUSER, before you can attempt to bring up a CICS region. On a production system the default user should *not* have access to any unnecessary CICS-supplied transactions. The resource access authorizations that you give to the default user should clearly be limited to those resources that you intend should be universally available, and therefore do not need to be restricted in any way. For information about defining the attributes of the default userid, see the *CICS/ESA CICS-RACF Security Guide*.

6 The dynamic transaction routing program, WLMDYP, is to be used for routing transactions that are defined with the DYNAMIC attribute.

7 This specifies the message text that is to be displayed on the screen by the CSGM (good morning) transaction when a terminal is logged on to CICS through VTAM or by the CESN transaction if used to sign on to CICS.

8 The resource group lists to be loaded when CICS starts up. The first resource group list shows the use of the *CTGI* naming convention. The groups of resource definitions included in these group lists are installed only during a cold start; in a warm restart the resource definitions still installed at shutdown are reinstated from the CICS global catalog.

9 Interregion communication is started during system initialization (IRCSTRT=YES), and the the CICS programs required are made available (ISC=YES).

10 The program autoinstall function is enabled at startup.

11 CICS allocates the read-only DSAs, RDSA and ERDSA, from read-only key-0 protected storage.

12 Storage protection is active (is enabled and CICS runs on hardware and an MVS release that supports storage protection).

13 CICS obtains the storage for the terminal control user area (TCTUA) in CICS key. This means that only programs executing in CICS key can modify the TCTUA, and user-key programs have read-only access.

14 The DFHSM0122 message informs you of the limit of DSA storage areas below the 16MB boundary. The DFHSM0123 message informs you of the limit of DSA storage areas above the 16MB boundary. These limits are set by the DSALIM and EDSALIM system initialization parameters respectively. For information about these storage areas, see the *CICS/ESA System Definition Guide*.

15 CICS storage protection is active without transaction isolation (STGPROT=YES and TRANISO=NO (the default) were specified). All storage in the CICS address space is addressable as in earlier releases.

16 If you specify MCT=NO (default) in the system initialization table or in the SIT overrides, the CICS monitoring domain builds a default monitoring control table. This ensures that default monitoring control table entries are always available for use when monitoring is active. The following message indicates that monitoring is inactive.

Chapter 3. Operating CICS in a multiregion environment

This chapter

outlines the operation of CICS regions in a multiregion environment using the multiregion operation (MRO) or intersystem communication (ISC) to communicate between CICS regions.

It assumes that you have already defined your multisystem environment, and are familiar with the concepts of CICS intercommunication facilities, as described in the *CICS/ESA Intercommunication Guide*.

You should consider using CICSplex System Manager/ESA (CICSplex SM) to manage multiple CICS regions. Those CICS regions can be:

- Independent, full-function CICS regions running on one or more connected CPCs¹
- Multiple, interconnected CICS regions functioning as a CICSplex, also on one or more connected CPCs.

For information about CICSplex SM, see the *CICSplex SM Concepts and Planning* manual, SC33-0786.

To provide the optimum transaction processing environment for your online business applications, you may choose to spread the workload of your business applications across several CICS regions. You can exploit the CICS intercommunication facilities to communicate between the CICS regions, and from CICS to other types of systems (such as IMS).

The CICS intercommunication facilities that you can use are:

- **Multiregion operation (MRO).** This is communication between two or more CICS regions running in the same MVS image, or separate MVS images within the same sysplex, using CICS internal facilities and protocols. The CICS MRO interregion communication facility is entirely independent of the SNA access method.
- **Intersystem communication (ISC).** This is communication between two or more systems in the same host, or in different hosts, through an SNA access method such as ACF/VTAM.

Using ISC, the hosts can be different operating systems, and the communicating systems do not both have to be CICS. For example, a CICS/ESA region running in an MVS image can communicate with a CICS/VSE partition running under VSE/ESA. Alternatively, on two MVS platforms, the communicating systems could be CICS/ESA at one end and IMS at the other.

¹ CPC. One physical processing system, such as the whole of an ES/9000 9021 Model 820, or one physical partition of such a machine. A physical processing system consists of main storage, and one or more central processing units (CPUs), time-of-day (TOD) clocks, and channels, which are in a single configuration. A CPC also includes channel subsystems, service processors, and expanded storage, where installed.

Using CICS intercommunication facilities, CICS functions can be separated into individual regions, the different types of CICS region being classified as resource managers. These CICS resource managers can reside in one or more MVS images. However, if they are in different MVS sysplexes, communication must be ISC. CICS regions in the same MVS image (or the same sysplex) can use MRO, which is the preferred method for performance reasons.

CICS regions segregated into resource-manager regions are generally known by the name of the principal resource they “own”. The names used in this book are summarized in “Notes on terminology” on page x.

The *CICS/ESA Intercommunication Guide* contains information about CICS intercommunication, and how to implement both intersystem communication and multiregion operation. This chapter assumes that you are familiar with the concepts of CICS intercommunication facilities, as described in the *CICS/ESA Intercommunication Guide*.

For information about MRO and its facilities, see the *CICS/ESA Intercommunication Guide*.

Enabling MRO

To be able to use CICS MRO, you must first install support for MRO, as described in the *CICS/ESA Intercommunication Guide*. In particular, when starting up your CICS regions, you must specify the `ISC=YES` system initialization parameter, to include the CICS programs required for MRO into your CICS regions.

Opening interregion communication (IRC)

Before a CICS region can use MRO to communicate with other CICS regions, each participating region must open interregion communication (IRC). You can do this in either of the following ways:

- Specify the `IRCSTRT=YES` system initialization parameter when you start the CICS region
- Use the `CEMT SET IRC OPEN` command while CICS is running.

Defining MRO connections

MRO connections for each CICS region are defined in the CICS system definition (CSD) file, by the `CONNECTION` and `SESSIONS` resource definitions. The definitions needed for a CICS region to communicate with other CICS regions are normally installed when you startup the CICS region.

Adding new MRO connections while CICS is running

While your CICS regions are running, you can add new connections to a CICS region by using the `CEDA INSTALL` command on that CICS region, to install the group containing the associated `CONNECTION` and `SESSIONS` resource definitions. For example:

```
CEDA INSTALL GROUP(CONNHT#1)
```


Installs the resource group, CONNHT#1, containing the CONNECTION and SESSIONS resource definitions for the TOR, CICSHTH1.

Notes:

1. You **do not** need to close interregion communication to be able to install new MRO connections.
2. CICS commits the installation of connection definitions at the group level only if all the connections in the group are installed successfully. If the install of any connection fails, CICS backs out the installation of all connections in the group. Therefore, when adding new connections to a CICS region while IRC is open, you must:
 - Ensure that the new connections are in a group of their own.
 - Install the group.

Changing MRO connections while CICS is running

While your CICS regions are running, you can change existing CONNECTION and SESSIONS definitions, by using the CEDA INSTALL command. However, you must first close IRC by using the CEMT SET IRC CLOSED command.

Closing interregion communication (IRC)

You can close interregion communication (IRC) by using the CEMT SET IRC CLOSED command while CICS is running.

Note: Before IRC can close, all pipes (sessions) of the external call interface must have been closed by the batch programs. For information about the external CICS interface, see the *CICS/ESA External CICS Interface*.

Chapter 4. CICS in an IMS data-sharing environment

This chapter is an introduction to operating CICS regions, generated with CICS local DL/I support, in an IMS data-sharing environment. It relates only to those CICS regions that are running with DLI=YES specified as a system initialization parameter. For information about accessing IMS databases using database control (DBCTL), see the *CICS/ESA CICS-IMS Database Control Guide*.

This chapter covers the following subjects about operating CICS regions in an IMS data-sharing environment:

page	Subject
29	Overview of IMS data-sharing, describing what IMS data-sharing is, and why you might want to use this facility.
31	Description of the four levels of IMS data-sharing.
31	Description of IMS database recovery control (DBRC), which controls backup, recovery, and access for all DL/I databases that are registered to it.
34	Overview of the internal resource lock manager (IRLM), which is essential for block-level data sharing.
34	Summary of the CICS system initialization parameters that you must code for IMS data-sharing.
34	Figures illustrating some sample IMS data-sharing configurations.
37	Description of operating CICS in an IMS data-sharing environment.
42	Description of the procedures (some given as flowcharts) that you can follow when using CICS in an IMS data-sharing environment. A list of these procedures is given on page 43.

An overview of IMS data-sharing

The following sections describe IMS data-sharing, the benefits of using it, and the different levels of data-sharing that you can use.

What is IMS data-sharing?

IMS data-sharing is a facility that allows more than one subsystem to have concurrent access to a DL/I database. Each subsystem can be:

- A CICS region running with CICS local DL/I support
- An IMS data communication (DC) online region
- An IMS database (DB) batch region
- A DBCTL region.

The subsystems can comprise any mixture of the possible types; for example, all the subsystems can be CICS regions.

All the subsystems can reside in a single host processor, or, depending on the level of sharing, in **physically adjacent** processors, as follows:

- Two host processors can participate in interhost block-level data-sharing, but they must be connected by a channel-to-channel link (CTC) or by a 37x5 control unit with a multiple-channel adapter.
- Up to four host processors can participate in the database level of sharing.

Access to the data is controlled by database recovery control (DBRC) through the authorization process described under “Access authorization” on page 32.

Why use data-sharing?

In an MRO environment, more than one CICS region can access the same database, but, except for the system that owns the database, the access is indirect. If a CICS region wants to access a database it does not own, it must ask the owning system to perform the access for it. The request can be made in one of three ways:

- **Transaction routing.** A transaction in the owning system is started from a terminal attached to the non-owning system.
- **Function shipping.** A transaction in the non-owning system transmits a specific access request to the owning system.
- **Distributed transaction processing.** A transaction in the non-owning system starts a transaction in the owning system and subsequently communicates with it.

Similarly, in a **CICS shared database** environment, a batch shared-database region can access a database owned by, and under the control of, a CICS region.

IMS data-sharing, in which CICS regions can participate, also allows more than one system to access the same database. The essential difference between IMS data-sharing and all the facilities described above is that, in an IMS data-sharing environment, each subsystem accesses the database **directly**, and no one subsystem owns the database. This eliminates the overhead of communication between CICS regions, at the cost of using the IRLM and DBRC.

IMS data-sharing gives users who are accessing databases through the CICS local DL/I interface certain facilities that are not otherwise available with that interface.

The use of IMS data-sharing:

- Is the only way that CICS local DL/I systems can share access to a database with an IMS DC region
- Allows the sharing of access to a database across host processors between multiple CICS local DL/I, IMS systems, or both
- Avoids the restrictions of the CICS shared database environment.

Data sharing between an IMS batch region and a CICS local DL/I system allows you to use the full functions of IMS batch programs; the restrictions associated with CICS shared database support do not apply. For information about these restrictions, see the *IMS Application Programming for CICS Users* manual.

The levels of IMS data-sharing

There are four levels of IMS data-sharing:

EXCLUSIVE LEVEL(Sharelevel=0)

Only one subsystem at a time can access the database. This level is intended to be used when you do *not* want access to a database to be shared.

DATA BASE LEVEL(Sharelevel=1)

Multiple subsystems can have read access.

One subsystem can update the database, provided all the other subsystems currently accessing the database have read-only (read without integrity) access. While authorized, the updating subsystem has exclusive updating access to the *entire* database. This access precludes the granting of update or read (with integrity) access to any other subsystem. Update access is not granted to a subsystem if any other subsystem is already authorized for update or read with integrity.

The duration of an authorization is from first schedule until normal subsystem shutdown (or earlier, using the CEMT SET DLIDATABASE command).

There is a maximum of four host processors.

INTRAHOST BLOCK LEVEL (Sharelevel=2)

Multiple subsystems can have access (update or read).

A subsystem authorized for update is granted sole use of a **physical block** of data. While it has use of that block, access is not granted to any other subsystem that is authorized for update or read with integrity. The remainder of the database is available to other subsystems.

An authorization starts with the scheduling call and ends when the local subsystem terminates or issues a RECOVERDB LOCAL command, or when any subsystem issues a RECOVERDB GLOBAL command.

All subsystems are in one host.

INTERHOST BLOCK LEVEL (Sharelevel=3)

As Sharelevel=2, except that the subsystems can be distributed between two host processors.

Data sharing is implemented through two IMS facilities:

1. IMS Database Recovery Control (DBRC)
2. IMS Resource Lock Manager (IRLM).

DBRC is a prerequisite for all levels of data-sharing, and the IRLM is needed for block-level data-sharing.

Database recovery control (DBRC)

DBRC runs in the address space of each CICS local DL/I subsystem. DBRC controls backup, recovery, and access for all DL/I databases that are registered to it. These functions are essential to data-sharing, but DBRC can also be used in a non-data-sharing environment.

DBRC automatically records recovery data on a duplicated recovery control data set, known as the RECON data set. It also generates and validates JCL for backup and recovery jobs using IMS database utilities.

Registration of databases and subsystem sign-on

All databases that participate in data-sharing must be **registered** to DBRC by name and sharelevel. This is done by using the DBRC batch utility, DSPURX00. You need to repeat this job only if the recovery control data set is corrupted.

Each subsystem must **sign on** to enable DBRC to record the name of each active subsystem. For a CICS subsystem using the CICS local DL/I interface, signing on is automatic at system initialization time if DLDBRC=YES is specified in the system initialization table (SIT). The name of the CICS subsystem is the value of the APPLID parameter.

Access authorization

When a subsystem first attempts to access a given database, DBRC must **authorize** the access attempt before the subsystem can successfully access the database. The access intent is that specified in the ACCESS operand of the DFHDLDBD macro defining the database. This must not be more stringent than that defined in the program specification block (PSB). The possible values of the ACCESS operand, in order of stringency, are RO (read-only, read without integrity), RD (read with integrity), UP (update), and EX (exclusive).

Authorization is granted if no conflict is detected. Authorization is denied if the database is already authorized to another subsystem at a conflicting level, or if an error condition exists. For a full list of the possible error conditions, see the *IMS Messages and Codes manual* (in the explanation of the reason codes associated with message DFS047A).

Table 2 on page 33 illustrates the authorization rules by showing examples of authorization requests and their results. In each example, the result is the same if either subsystem is replaced by an IMS subsystem with equivalent access requirements (see Table 3 on page 33).

Subsystem termination and canceling access authorization

If a CICS (or any) subsystem terminates normally, DBRC cancels the authorization for the databases to which it had access. If a subsystem terminates abnormally, DBRC records the subsystem failure.

If a failing subsystem has updated databases, DBRC retains any authorizations held on its behalf, and the IRLM retains the locks on any databases and physical blocks of which it had control at the time of failure. This allows the subsystem, on emergency restart, to back out the effects of uncompleted transactions and restore the integrity of the data.

Task failure within a subsystem

If a CICS (or any subsystem) task fails, any authorizations or locks are held, permitting dynamic transaction backout.

If SHARELEVEL=2|3, the locks are released at the next syncpoint, which occurs either before transaction restart or during task termination. This is simply a special case of the application of the general rule.

Failure of backout during emergency restart or dynamic transaction backout

If backout failure occurs, **all** databases being updated by the affected transaction are stopped, and the subsystem takes all actions to preserve data integrity as described in the *CICS/ESA Recovery and Restart Guide*. Similar action is taken in all other subsystems (they are advised by the IRLM), and DBRC prevents further authorization of these databases.

The IMS batch utilities can be run in another region to restore the databases.

Table 2. Examples of authorization requests to DBRC in an IMS data-sharing environment, and their results

Share Level	Subsystem Running (database already authorized)	Subsystem Starting (attempting to obtain authorization)	Result
0	CICS ACCESS=EX	CICS ACCESS=RD	Unsuccessful
0	CICS ACCESS=RD	CICS ACCESS=EX	Unsuccessful
0	CICS ACCESS=RO	CICS ACCESS=RD	Unsuccessful
1	CICS ACCESS=RO	CICS ACCESS=RD	Successful
1	CICS ACCESS=RD	CICS ACCESS=EX	Unsuccessful
1	CICS ACCESS=UP	CICS ACCESS=RO	Successful
1	CICS ACCESS=UP	CICS ACCESS=RD	Unsuccessful
1	CICS ACCESS=UP	CICS ACCESS=UP	Unsuccessful
2 or 3	CICS ACCESS=RD	CICS ACCESS=RD	Successful
2 or 3	CICS ACCESS=RD	CICS ACCESS=UP	Successful
2 or 3	CICS ACCESS=UP	CICS ACCESS=UP	Successful
2 or 3	CICS ACCESS=EX	CICS ACCESS=RD	Unsuccessful

Table 3. IMS data-sharing subsystems access levels (compared with the equivalent CICS subsystems access levels)

CICS subsystem	IMS BATCH subsystem	IMS DC subsystem
ACCESS operand of DFHDLDBD macro	PROCOPT value in PSB	ACCESS keyword of DATABASE system definition macro
RO	GO	RO
RD	G	RD
UP	I, R, D, or A	UP
EX	L or E	EX

Internal resource lock manager (IRLM)

The internal resource lock manager (IRLM) is a stand-alone resource-locking facility that runs in its own region in each host in an IMS data-sharing configuration. The IRLM is essential for block-level data-sharing. The IRLM does the following:

- Takes over as lock manager from the program isolation lock manager.
- Maintains the integrity of data-shared between multiple subsystems.
- Acts as a vehicle for the notification of events such as data set extensions, buffer invalidation, and database backout failures.
- Provides support for global commands.

Coding CICS system initialization parameters for IMS data-sharing

The following is a summary of the CICS system initialization parameters that you must code to use IMS data-sharing:

DLI|DL1

You must code DLI|DL1=YES if you want to use CICS local DL/I support.

DLDBRC

You must code DLDBRC=YES as a SIT parameter if you want to run CICS with the local DL/I interface and use IMS data-sharing. DLDBRC=YES can be coded only in the SIT; it is not available as a SIT override.

DLIRLM

You can run CICS with IMS block-level data-sharing. To do so, you must specify a DLIRLM=YES|*name* system initialization parameter to specify the name of the IRLM. If you code DLIRLM=YES, the default name IRLM is assumed.

You can run CICS with XRF. If you do, you can use the same SIT for the active and alternate CICS regions, but different IRLMs if the active and alternate are running in different MVS images. In this case you must provide the name of the IRLM as a SIT override.

For information about coding system initialization parameters, see the *CICS/ESA System Definition Guide*.

Sample IMS data-sharing configurations

Figure 4 through Figure 7 (on page 35 through page 36) illustrate some examples of data sharing configurations. These figures do not represent every possible situation, but show the general principles that apply to all configurations.

For more examples of IMS data-sharing configurations, see the *IMS System Administration Guide*.

Note: The letters CTC that appear in some of the figures mean channel-to-channel connector.

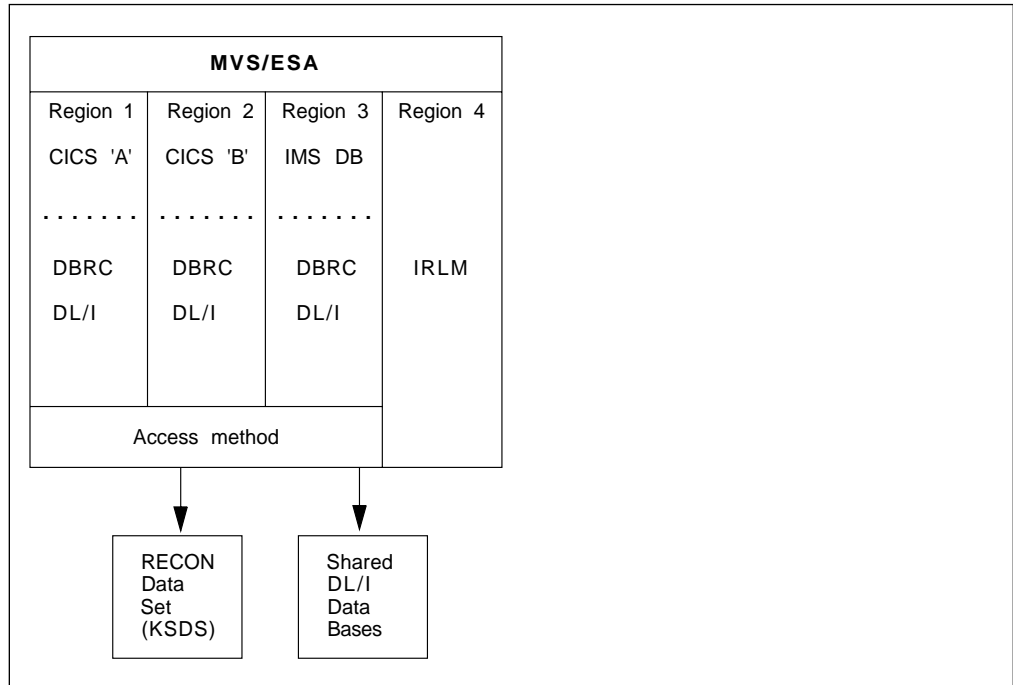


Figure 4. Sample IMS data-sharing: configuration 1. Intrahost data-sharing between CICS and IMS DB systems.

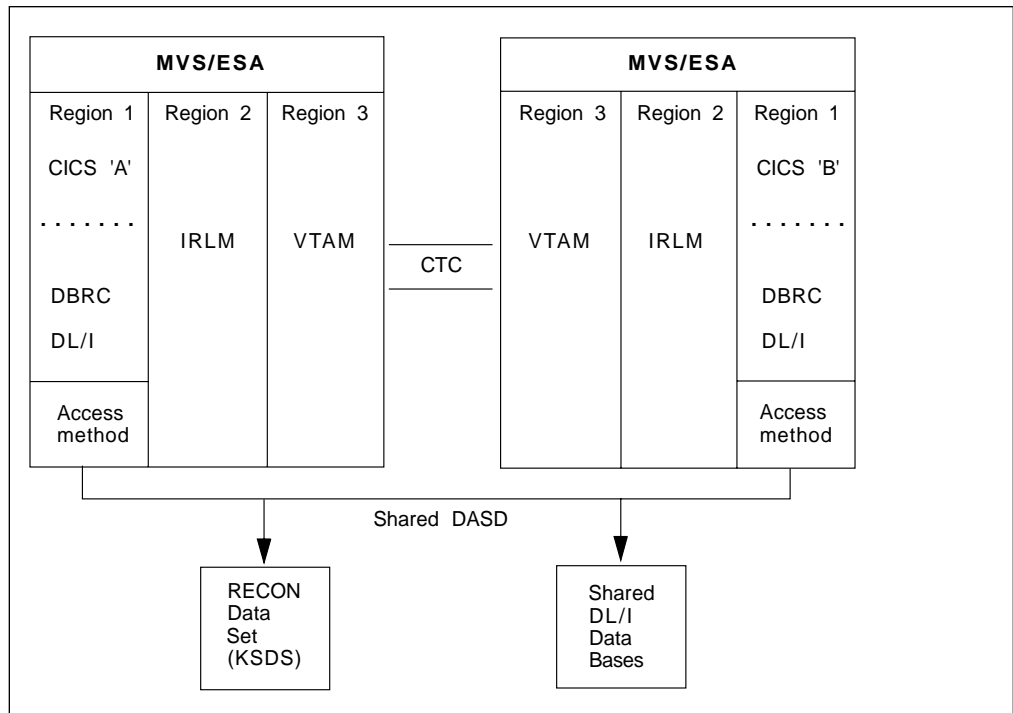


Figure 5. Sample IMS data-sharing: configuration 2. Interhost data-sharing between CICS regions.

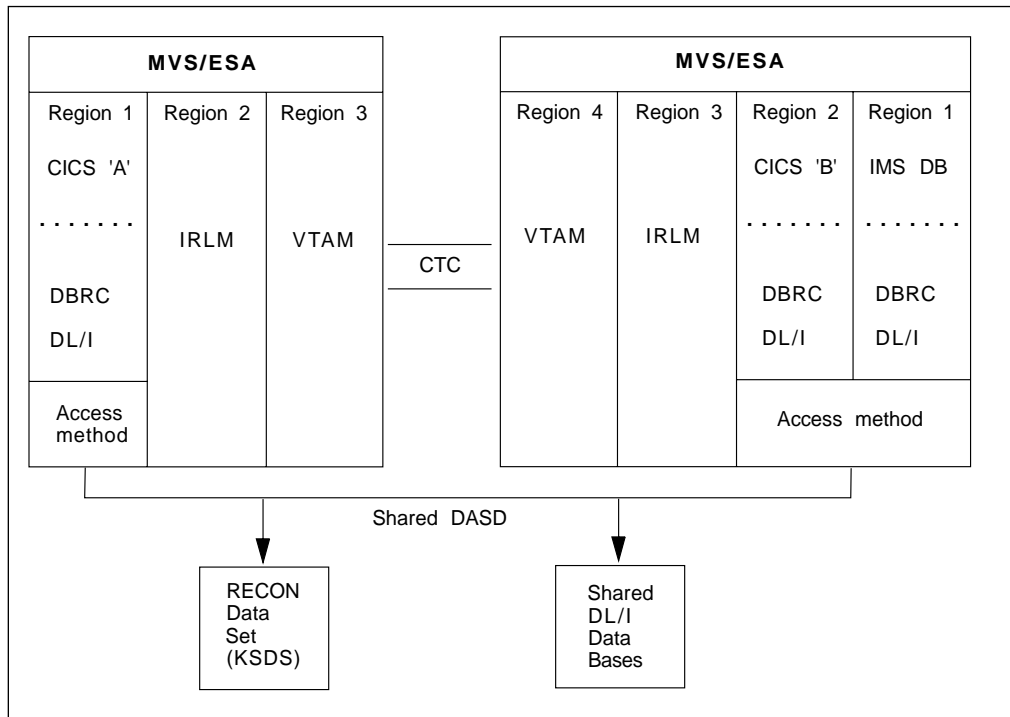


Figure 6. Sample IMS data-sharing: configuration 3. Interhost data-sharing between CICS and IMS DB systems.

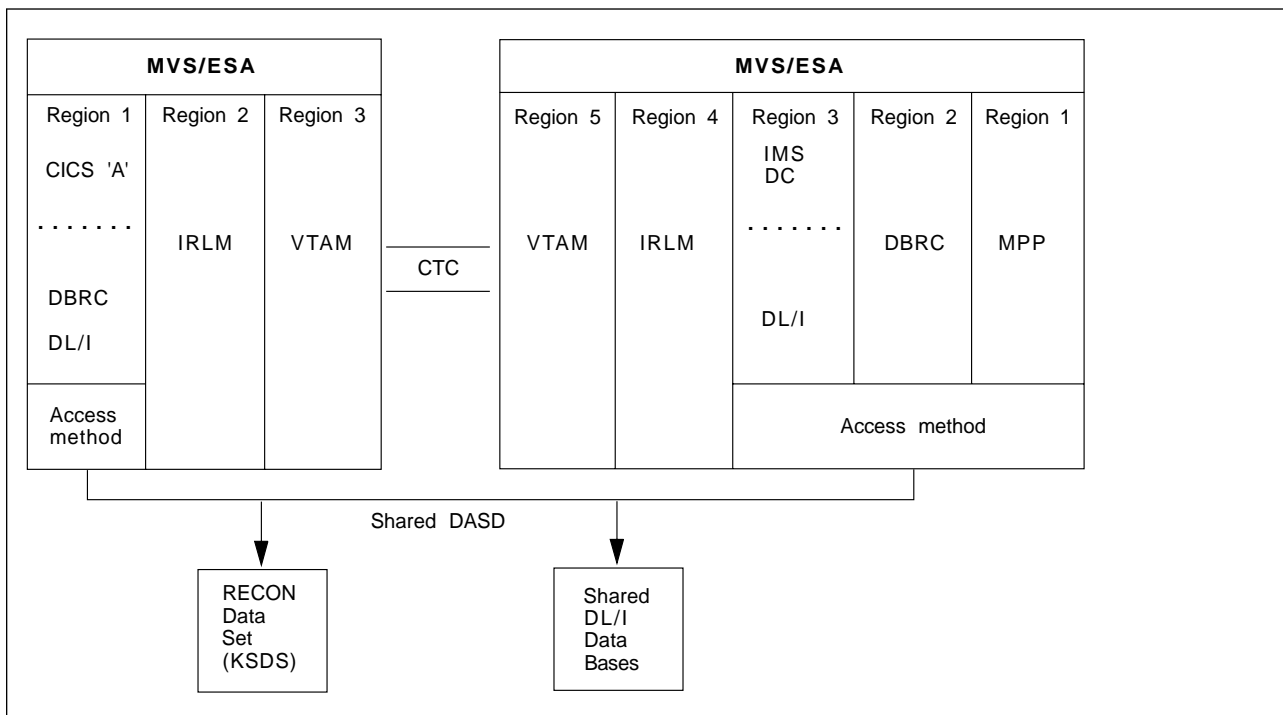


Figure 7. Sample IMS data-sharing: configuration 4. Interhost data-sharing between CICS and IMS DC systems.

Overview of operating CICS in an IMS data-sharing environment

Operating CICS when it is running as a subsystem in an IMS data-sharing environment is similar to normal CICS operation. This section describes the main points of difference. “Operating procedures and flowcharts” on page 42 describes procedures that you can follow when operating CICS in an IMS data-sharing environment, and gives some flowcharts that you can follow.

Start up the IRLM

You must start up the IRLM as well as CICS. Ideally, start up the IRLM first. Otherwise, you are prompted to do so during startup, and CICS has to wait while you start the IRLM.

XRF considerations

You can run CICS as a subsystem in an IMS data-sharing environment, with both:

- The CICS extended recovery facility (XRF=YES)
- Interhost block-level (sharelevel=3).

To do so, you must run a special type of DL/I batch job that always remains ready and identified to the IRLM on the MVS image on which the alternate is started. (This job is sometimes called the **ready region**.) The job should be running all the time the alternate CICS is in standby mode, to ensure that the IRLM running on the alternate’s MVS image retains any locks that are held at the point of failure of the active CICS.

This IMS batch job, which you define with a type code RDI in the PARM parameter on the EXEC statement, is necessary because the alternate CICS does not identify itself to the IRLM until it begins to takeover. If no other data-sharing partners are running on the same MVS image as the alternate CICS, and the active CICS fails, the IRLM flushes all subsystem information and releases any locks relating to the failed active CICS. By continuing to be identified to the IRLM, the IMS RDI job ensures that locks are retained (it appears to the IRLM and DBRC just like any DLI or database batch region). Retention of the locks through the existence of the RDI job means that the data-sharing availability improvements (DSAI) introduced in IMS Version 2 are available to the alternate CICS after it has successfully completed the takeover. After takeover, anything which was available to the failed CICS region is now available to the new active CICS region.

Note: There is no loss of database integrity without an IMS RDI job: the RDI job is there only to ensure that the alternate CICS can continue IMS data-sharing at block-level, as described.

Starting the IMS RDI (ready region) job on the alternate MVS image

The ready region is non-swappable, and always waits until:

- The IMS RDI job is canceled, or
- The IRLM to which it is identified fails. (IMS terminates the RDI job with abend 3303.)

The sample job in Figure 8 on page 38 shows you how to code the PARM parameter on the EXEC DFSRRC00 statement.

```

//IRLMRDI JOB 'accounting info',name,MSGCLASS=A
//RDI EXEC PGM=DFSRRCO0,REGION=0M,TIME=1440,
// PARM=(RDI,anyname,,,0000,,0,,N,0,T,,N,Y,Y,ir1name,N) 1
//STEPLIB DD DSN=IMSVS.RESLIB,DISP=SHR
//DFSRESLB DD DSN=IMSVS.RESLIB,DISP=SHR
//IEFRDER DD DUMMY 2
//SYSUDUMP DD SYSOUT=A
//RECON1 DD DSN=IMSVS.RECON1,DISP=SHR 3
//RECON2 DD DSN=IMSVS.RECON2,DISP=SHR
//

```

Figure 8. Sample job to start the IRLM ready region

Notes:

1 *anyname* is a dummy program name, and *ir1name* is the four-character name of the IRLM associated with the RDI job.

2 You can define IEFRDER as a dummy DD statement.

3 You must include the DBRC RECON data sets.

Note: The IMS PSB and DBD data sets are not required.

Messages

You see some additional messages compared with a CICS non-data-sharing operating environment. The following table enables you to identify the source of each message by its initial identifier.

Source	Initial identifier of message
CICS	DFH
IMS	DFS
DBRC	DSP
IRLM	DXR
MVS	lxx (xx varies)

In a data-sharing environment, and knowing the **source** of a message, you need to know the **subsystem** it refers to.

IMS messages are always followed by the subsystem identifier. For a CICS subsystem, you code the subsystem identifier on the SYSIDNT parameter in the SIT (default “CICS”). If you specify SYSIDNT=CICS, it appears in messages as CICA.

The CICS message, DFHSI1517 “Control is being given to CICS,” is prefixed by the APPLID parameter, like this:

```
DFHSI1517 - DBDCCICS: Control is being given to CICS
```

Note: If you are running CICS with XRF, it is the specific APPLID that is displayed in this CICS message.

The APPLID parameter is specified in the SIT, or as a system initialization override, and “DBDCCICS” is the default.

If neither a SYSIDNT nor an APPLID parameter is included in a message, you can use the MVS job number to identify the subsystem.

There are obvious advantages in making the APPLID and SYSIDNT parameters of the same CICS region as similar as possible. However, the maximum number of characters for these parameters is different, four for SYSIDNT and eight for APPLID. Also the APPLID value is constrained by the fact that it is used as the ACF/VTAM APPLID.

Copying disk logs

You must always copy disk logs before they are overwritten, to ensure that your CICS local DL/I databases can be forward-recovered. If you are using disk logging, you should define your system log with two data sets (JTYPE=DISK2), and also with one of the journal options that enables you to archive a log data set when it is full or when CICS has finished using it. You can choose from two of these options, which you code on the Jouropt parameter in the JCT entry:

AUTOARCH This operand (recommended) causes CICS to archive a log data set automatically when it is full or when CICS has finished using it, using the skeletal JCL which you provide as a member of a PDS that is defined to CICS. A sample of the skeletal JCL is supplied, in member DFH\$ARCH in CICS410.SDFHINST, for this purpose. For details of the DFHJPDS data set, and its use in automatic journal archiving, see the *CICS/ESA System Definition Guide*.

PAUSE This enables you to take an archive copy of the inactive data set each time that a log switch occurs. CICS prompts you with a WTOR console message when a data set is full, and does not reuse the data set until you reply to the outstanding message that the archive has been done successfully.

Code only one of these options on the Jouropt parameter.

In a DBRC environment you must use the IMS archive utility, DFSUARC0, to take the copy, so that the data set produced is known to DBRC and can be used if recovery becomes necessary.

Note: You cannot use DFSUARC0 directly after a CICS or MVS failure if no stop time has been recorded for the disk log, or the system-log data set has not been physically closed. First restart CICS with START=LOGTERM (for information about this special CICS startup option, see the *CICS/ESA System Definition Guide*), or perform an emergency restart.

You cannot use the DBRC GENJCL command to generate the JCL for DFSUARC0. You must create your own job stream and modify it each time you use it, as both the input and output data sets vary. Samples of the JCL required to run DFSUARC0 are given in Figure 9 on page 40. You see that these jobs are shown with DD statements for the RECON data sets. However, these are required only if dynamic allocation is *not* used, and dynamic allocation of the RECON data sets is the recommended method.

```

//SAMPLE JOB
//*
//* Sample JOB for DFSUARC0
//*
//LOGCLO EXEC PGM=DFSUARC0,
//      PARM='CICS=Y,DBRC=Y'
//*
//STEPLIB DD DSN=IMSVS.RESLIB,DISP=SHR
//*
//* CICS disk log data set
//*
//DFSSLDSP DD DSN=CICS410.DFHJ01A,DISP=SHR,DCB=RECFM=VB
//*
//* Tape archive volume
//*
//DFSSLOGP DD DSN=LOG1,LABEL=(,SL),UNIT=TAPE,
//      VOL=SER=valid,DISP=(NEW,KEEP)
//SYSIN   DD DUMMY
//*
//RECON1   DD DSN=RECON1,DISP=SHR
//RECON2   DD DSN=RECON2,DISP=SHR
//RECON3   DD DSN=RECON3,DISP=SHR
//SYSUDUMP DD DSN=SYSOUT=A
//SYSPRINT DD DSN=SYSOUT=A

```

Figure 9. DFSUARC0 sample JCL

Image copies

At intervals which you must determine, you should take an image copy of each database, to be used as the basis of any forward recovery of the database. Use the IMS utility, DFSUDMPO, to take the image copy: for information about using DFHSUDMPO, see the *IMS Utilities Reference manual*.

For CICS local DL/I, you can use the CBRC GENJCL command to generate the JCL for this job.

For information about the /RMxxxx commands if you are using DBCTL, see the *IMS Operator's Reference manual*.

Change accumulation

The DBRC change accumulation utility, DFSCYM0, takes as input your archived logs, and produces as output a change accumulation log containing the minimum entries needed for forward recovery of a specified database. For information about DFSCYM0, see the *IMS Utilities Reference manual*.

After running DFSCYM0, you can delete the original log entries for the database by using the CBRC command DELETE.LOG INACTIVE.

To release your archived log volumes or data sets, you should perform change accumulation for all your databases on a regular basis. You should also normally perform change accumulation before recovering a database.

Forward recovery

When forward recovery of a database is necessary, use the IMS database recovery utility, DFSURDB0; for information about DFSURDB0, see the *IMS Utilities Reference manual*.

You can use the CBRC GENJCL command to generate the JCL for this job.

DFSURDB0 takes as input a specified image copy and the appropriate change accumulation log. Change accumulation is forced if necessary.

Database integrity

A database that is being shared has a JCL disposition of SHR and is defined with a VSAM cross-system SHAREOPTION of 3, which permits full sharing. Therefore, the database can be freely accessed by any region that is not signed on to DBRC. Your operating procedures must ensure that, while a database is being shared, it is never **updated** by a region that is not signed on to DBRC. Note also that if such a region reads the database, the reads lack integrity.

Every time a backout failure occurs, you must run the IMS batch backout utility for the PSB. If you do not do this, any attempt to schedule an affected database is unsuccessful. It is easy to forget this when a backout failure occurs during an emergency restart, which itself then fails. In this case, remember to run the backout utility after a later successful emergency restart, to reset the backout failure flag in the RECON data set.

CICS-supplied transactions

The CBRC transaction enables you to issue DBRC commands from a CICS subsystem. If you use this transaction, you must allow for 450KB of operating system storage outside the CICS dynamic storage area (DSA). If you cannot afford this virtual storage overhead, note that the DBRC commands can also be issued by running the DBRC batch utility. For information about storage parameters, and for an outline of CICS storage requirements, see the *CICS/ESA System Definition Guide*.

The CEMT transaction can be used to control CICS local DL/I databases, and the CEMT INQUIRE|SET DLIDATABASE(fileid) command has two operands especially for data-sharing:

ACCESS to specify the access intent of CICS against a DL/I database

GLOBAL to transmit the command to all data-sharing subsystems in addition to processing it on the local CICS region.

For information about terminal operations for DBRC and data sharing, see the *CICS/ESA CICS-Supplied Transactions* manual.

Naming multiple CICS regions

Your operating procedures should ensure that it is not possible to bring up two CICS subsystems with the same APPLID. If this happens, two subsystems are sharing the same SUBSYSTEM record in the RECON data set, and data integrity is lost.

Operating procedures and flowcharts

This section provides an outline description of the procedures to follow in starting and running the online CICS region, when it is functioning as a subsystem in an IMS data-sharing environment.

Some of the procedures are given as flowcharts. These flowcharts are not exhaustive; they only provide a basis for indicating the operating considerations when using CICS as a data sharing subsystem with disk or tape logging.

Appendix A, "Operating procedures" on page 291 contains definitions of the departmental/job titles, forms, and tables that are referred to in some of these charts and their accompanying notes.

How to use the flowcharts

The flowcharts all have the same format:

- A block diagram shows the actions and decisions to be made. If a block refers to another chart, you should turn to that chart for more details.
- Where appropriate, the actual commands entered and the responses received, appear next to the block diagram. The notation used for these commands and responses is as follows:

sc-e	Command entered at MVS console
sc-r	Response received at MVS console
mt-e	Command entered at master terminal
mt-r	Response received at master terminal.

- Sometimes, additional notes about the action or decision blocks are provided on the page facing the flowchart (indicated by a number on the block diagram). You probably need to refer to these notes only when you are unfamiliar with the procedure.

List of procedures

<i>Procedure</i>	<i>see page</i>
OVERVIEW OF ONLINE PROCEDURES	
Sequence of operations	43
CICS STARTUP PROCEDURES FOR LOCAL DL/I	
Start up CICS	45
A-3 CICS initialization with local DL/I	46
SYSTEM MONITORING PROCEDURES	
Normal CICS region monitoring	48
SHUTDOWN PROCEDURES	
A-4 Shut down CICS	51
ERROR PROCEDURES	
B-1 Action on user problem	54
B-2 IMS database backout failure	57
B-3 IMS database medium failure	60
B-4 IRLM connection failure	63
B-5 IRLM communications failure	66
B-6 Machine check, power failure, MVS loop, or abend	69
Errors that do not cause CICS to abend	71
Abend errors	73
EMERGENCY RESTART (DISK LOGS)	
Perform emergency restart	74
LOG I/O ERRORS	
C-1 Log errors	76

Sequence of operations

The following sequence highlights the main steps to be followed when starting up and shutting down a CICS local DL/I system that is using IMS data-sharing.

Prepare log sheet and log volumes

You need a CICS online log sheet for each startup operation. This helps ensure that disk logs have been copied and that tape logs are available.

You must keep the system log on disk or standard-labeled tape.

If you use standard-labeled tape logs, you need log tapes that have been allocated a volume identifier (the volume-serial number). This serial number is displayed by CICS during each startup.

Set up a procedure at your installation to control and maintain these log tapes.

Initialize the network

If data-sharing is at the interhost level, network initialization includes bringing up the CTC, or communication controller, links. It also includes such things as switching on modems and other necessary hardware, loading the communication controller control program, and performing VTAM initialization, if applicable.

Start up the IRLM

If data-sharing is at the block level, you must start up the IRLM as well as CICS. The IRLM runs in a separate address space, and can be started with a procedure using the START command or by submitting the startup job stream through the MVS input reader.

Start up CICS

CICS determines the type of startup to perform if you have chosen the START=AUTO option, using the control information recorded in the CICS global catalog.

If you are running CICS with the extended recovery facility (XRF), and using the same SIT for the active and alternate CICS regions, you may need to provide the IRLM name in the startup job stream. You can do this either in the PARM string on the EXEC statement, or in the SYSIN data set, but you cannot supply the IRLM name through the console.

Follow special instructions

For any session, you may have to follow special procedures when the system is initialized. For example, you may want to alter the initialization parameters or there may be a transaction that is to be stopped because it is giving trouble, or a special message may need to be broadcast to all users.

Complete log sheet for startup

You should fill in the following information on the CICS online log sheet at startup time:

- Start time.
- Your name.
- Start type.
- The label allocated by CICS to the log tapes used for output in this run. Optionally, you can fill in the volume serial numbers of the log tapes if standard-labeled tape logs are used.
- Comments and incidents (if any).

Archive disk log data sets

If you are using disk data sets for the system log, use the IMS log archive utility, DFSUARC0, to archive log data sets each time a log switch occurs.

If you have defined the journal option AUTOARCH on the Jouropt parameter, you should ensure that the skeletal JCL in the DFHJPDS data set specifies DFSUARC0 as the archive utility.

Shut down CICS

Complete the online log sheet indicating the nature of the shutdown; for example, state if it is successful with a warm keypoint.

If disk logging is used, use the IMS log archive utility, DFSUARC0, to take an archive copy of the last log data set used. If you are using automatic journal archiving (Jouropt=AUTOARCH) this is done automatically for you during the CICS termination process.

CICS startup procedures for local DL/I

This section illustrates some of the main points during the initialization of a CICS region started with DL/I=YES, DLIRLM=YES, and DLDBRC=YES.

Start up CICS

Start IRLM for block-level data-sharing

If you are using block-level data-sharing (share levels 2 or 3), you should start up the IRLM before you start up CICS.

Copying disk system-log data sets

If you are controlling system-log archiving manually, ensure that you have copied all disk system-log data sets for recovery purposes before you start CICS.

Alternatively, by coding the automatic journal archiving option in the JCT entry for the system log (JOUROPT=AUTOARCH), you can ensure that CICS takes care of the archiving for you. This is the recommended option which, briefly, works as follows:

- During normal shutdown, CICS submits automatically an archive job for the current system-log data set, and any other journal data sets that are defined on disk with JOUROPT=AUTOARCH.
- If CICS detects that a required journal data set is not ready during startup (because the journal was not archived during the previous shutdown because of an abnormal termination), CICS submits automatically the archive job during startup.
- Each time that a journal data set becomes full and the journal control program makes a data set switch, CICS submits automatically the job to archive the full data set.

The status of journal data sets defined with AUTOARCH is maintained on the journal archive control data set (JACD). Once the JACD is defined, do not modify it, except by using the supplied functions of the JACDU utility, because the status is crucial to the archiving process.

If you specify AUTOARCH when you are running CICS with local DL/I support, use DFSUARCO as the copy utility in the archiving job step. For more information about automatic journal archiving, see “The automatic journal archiving utility” on page 146.

For information about the DFSUARCO utility, see the *IMS Utilities Reference manual*.

Copying tape system-log data sets

Normally, it should not be necessary to copy tape logs unless they form part of a cyclic series and are likely to be overwritten. For information about the options available for tape logs, and the use of linear or cyclic series, see the *CICS/ESA Recovery and Restart Guide*. See also the DFHJCT macro instruction definition in the *CICS/ESA Resource Definition Guide* manual.

Start the CICS job (or task)

There are several ways to start up CICS depending on the system environment. You can start the CICS job from a procedure by using the START command, or you can submit the CICS job stream through the reader. For more information about the CICS system initialization process, see the flowchart A-3, on page 46.

Chart A-3: CICS initialization with local DL/I

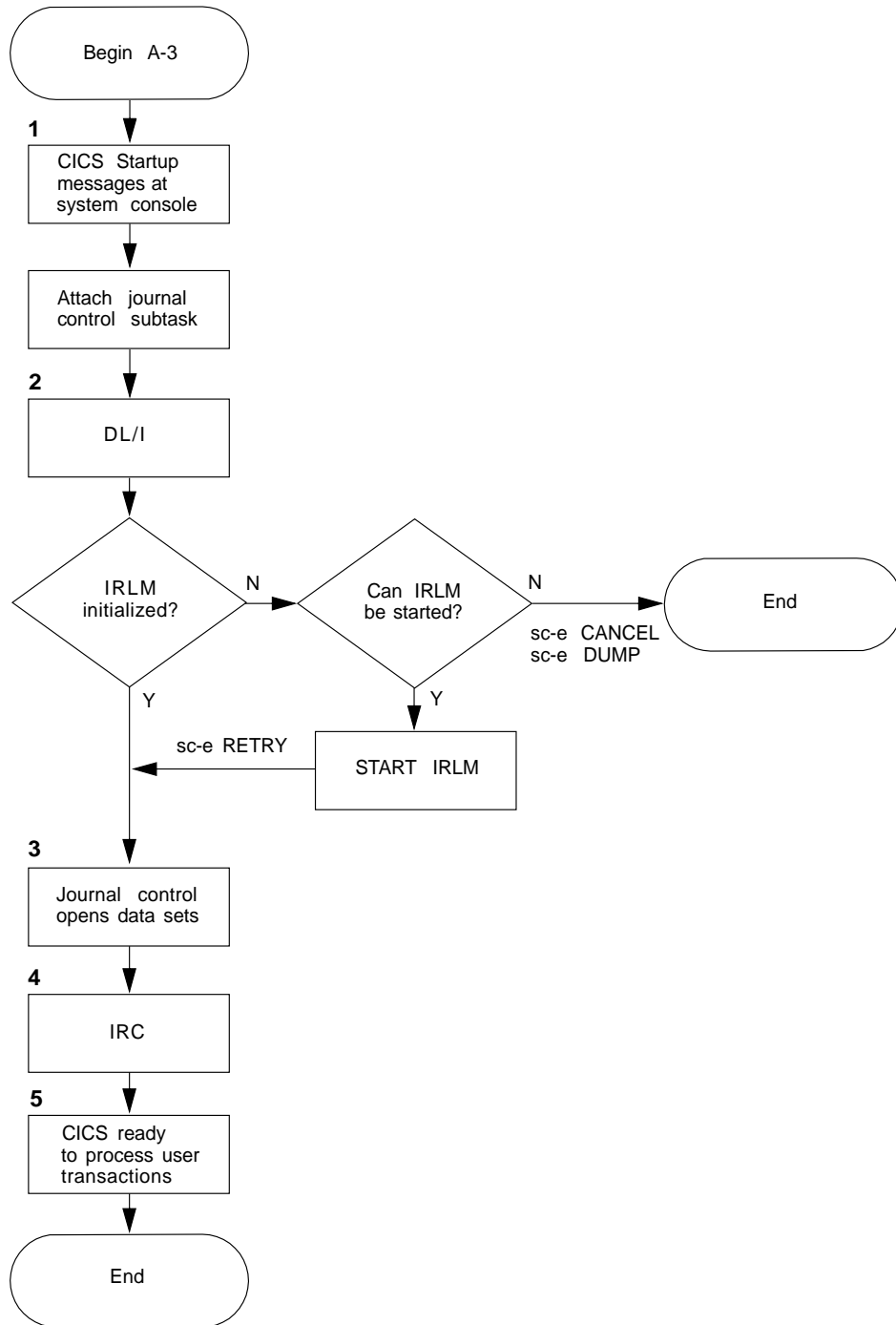


Figure 10. Chart A-3: CICS initialization with local DL/I

Notes for chart A-3

1. CICS startup messages at system console

CICS startup begins by displaying any SIT override parameters read from SYSIN.

If you have specified PARMERR=INTERACT, CICS prompts you to correct any errors it finds in SIT override parameters coded in the PARM field, the SYSIN data set, or entered through the console. Enter corrections through the console in response to the REPLID on the CICS DFHPA1912 message. However, for a production CICS region you should code PARMERR=ABEND to prevent any unauthorized or accidental modification of your CICS initialization parameters.

2. DL/I initialization IRLM error

If you have not initialized the IRLM, or you have initialized it with a name different from that specified in the SIT, the following messages appear:

```
DFS039I IRLM IDENTIFY REQUEST FAILED, RC=08-40 CICA
@nn DFS039A IRLM NOT ACTIVE. REPLY RETRY, CANCEL, OR DUMP. CICA
```

The default name is "IRLM," and this is assumed when you code DLIRLM=YES. The default name is quoted here within the DFS039A message, in the phrase "IRLM NOT ACTIVE." If possible, initialize the IRLM with the correct name, and reply RETRY to message DFS039A. For further information about these messages, see the *IMS Messages and Codes manual*.

3. Journal control opens data sets

You see the following journal control messages during startup:

```
DFHSI1510I applid Journal control subtask is being attached/entered
:
DFHSI1516I applid Opening journal data sets.
:
DFHJC4508 applid CICS SYSTEM LOG. PRIMARY DATA SET NOW RECEIVING OUTPUT
ON cuu
DFHJC4500 applid 01 of 01 journals successfully opened
```

If you are operating with your journals on standard-labeled tapes you see DFHJC4503 instead of DFHJC4508, and message DFHJC4502 appears first, requesting that the indicated volume ID (for standard-labeled tape journals) be mounted. The CICS-allocated label (date and time) is included as part of the message. For standard-labeled tape journals, the external recording of the CICS-allocated label is optional.

After a successful open for standard-labeled journals, message DFHJC4503 follows to indicate the label CICS has allocated to the tape volume. The label is in the form:

```
yyddd/mmm RUN hhmmss
```

where:

```
yyddd    = Today's operation date
mmm      = Volume sequence number
hhmmss  = Start time of CICS execution.
```

Prepare a physical label in this form for use later when the tape is unloaded.

For standard-labeled tapes, message DFHJC4503 also includes the volume ID.

If a labeled journal does not have enough volumes defined in its series, message DFHVC4710 appears, warning the operator that more volumes should be defined. If this warning is ignored and CICS runs out of tapes for this journal, message DFHVC4700 prompts the operator to supply a volume serial number to be used for the journal.

4. Interregion communication session started

If your system supports MRO or the CICS shared DL/I database facility, and has started the IRC session at initialization (IRCSTRT=YES is coded in the SIT), you see the following message:

```
DFHSI1519I applid The interregion communication session was successfully
started
```

5. CICS initialization complete

When CICS initialization is complete and CICS is ready to process user transactions, you see the following message:

```
DFHSI1517 applid Control is being given to CICS.
```

DBRC and IRLM warning messages: If CICS startup is normal you should not see the following messages:

```
DFHDL3914 applid Retained DBRC authorizations for datasharing users have
been released
DFHDL3916 applid Retained IRLM locks have been released.
```

These messages imply that the last run of this CICS region terminated abnormally, with DBRC authorizations and IRLM locks still in force. If this is the case, you should not be starting CICS with the START=COLD parameter; CICS should be performing an emergency restart. By not doing so, you have lost the ability to back out uncompleted LUWs (logical units of work). The integrity of the affected databases can no longer be guaranteed.

If these messages appear during a cold start, inform the database administrator immediately.

You should ensure that this never occurs by always using the system initialization parameter START=AUTO.

Normal CICS system monitoring using master terminal commands

Monitoring tasks

From time to time, you can use the CEMT INQUIRE TASK command to query the status of tasks in the CICS region. If you notice that a user task with the same task number is suspended every time you display the task status, you may suspect that the task is in a hard wait and you should refer the problem to DC Administration.

If a program is currently active in a shared database region, the mirror task is also displayed.

Monitoring transactions

CEMT INQUIRE TRANSACTION displays the status of all transactions. If any transaction has a status of "Dis" (disabled), and this is different from normal as defined by DC Administration on the "Transaction and program table" form, you may have to consult DC Administration.

Monitoring programs

CEMT INQUIRE PROGRAM displays the number of times the program is currently being used, and the number of times the program has been used. This enables you to see the load on each transaction. You may want to increase the priority of a frequently used transaction, or adjust the maximum number of tasks by class (CEMT INQUIRE TCLASS).

Also notice the “Ena” or “Dis” status of each program, and report to DC Administration if it is different from the associated resource definition in the CSD.

Monitoring partner programs

CEMT INQUIRE PARTNER displays the following information for all partner programs:

- the network name
- the symbolic destination name
- the name of the CICS profile for the partner
- the length of the transaction program on the partner LU
- either the name of the transaction program on the partner LU or the alternative transaction identification (in hexadecimal).

Tape journal volume management

CEMT INQUIRE VOLUME JOURNAL(x) displays the status of the volumes in a journal series. CEMT INQUIRE JOURNALNUM displays the status of a journal (input or output). The CEMT SET VOLUME command allows you to manage this resource.

Disk journal management

The messages that are issued depend on whether you are using disk logging with the JOUROPT=PAUSE or JOUROPT=AUTOARCH option in the DFHJCT TYPE=ENTRY macro instruction for the system log.

JOUROPT=PAUSE

The message DFHJC4583 is issued when CICS is ready to:

- Switch to the second data set (if you specified JTYPE=DISK2); or
- Close and restart at the beginning of the data set (if you specified JTYPE=DISK1). For single data set logs, message DFHJC4583 is followed by DFHJC4584. (However, JTYPE=DISK2 is the recommended option for system logs on disk; you should not be using a single disk data set for the system log unless you have a very special reason.)

In response to message DFHJC4583, use the IMS log archive utility, DFSUARC0, to archive the system log data.

JOUROPT=AUTOARCH

If a journal data set becomes full, CICS opens another data set and issues message DFHJC4508. This is followed by message DFHJC4553I when CICS submits the automatic journal archive job for the data set that is full.

Monitoring DL/I databases

CEMT INQUIRE DLIDATABASE(*fileid*) displays the status of a database, where *fileid* is the name defined to DL/I (in the DDIR control block). Make sure that all databases in the system are opened, and report to DC Administration if you find that a database has been stopped by the system.

Monitoring with DBRC commands

The CBRC transaction allows you to enter DBRC commands online. This enables you to interrogate and modify the RECON data set, which shows the status of subsystems, logs, and databases. For information about the CBRC transaction, see the *CICS/ESA CICS-Supplied Transactions* manual.

Monitoring IRLM

You can examine the status of the IRLM using the MVS modify command:

```
F irlmproc,STATUS[,keyword]
```

For information about the MODIFY (F) irlmproc command, see the *IMS Operator's Reference* manual.

Note: The CICS region is usually monitored by the CICS master terminal operator or the MVS console operator.

Chart A-4: Shut down CICS

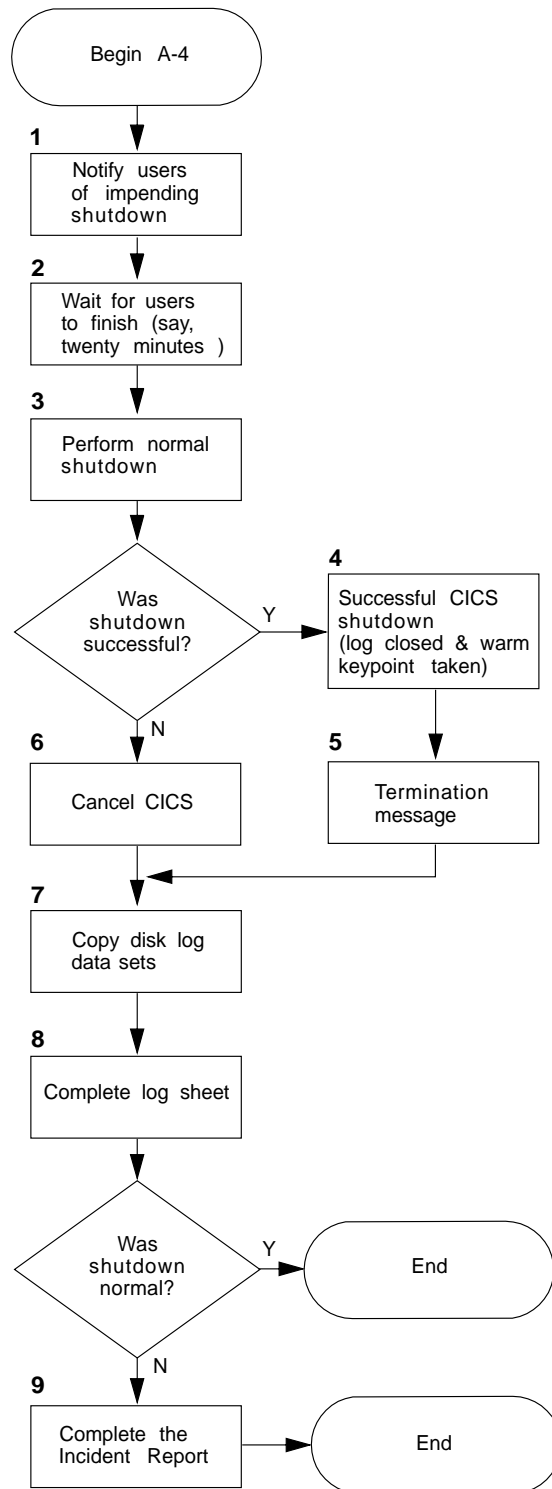


Figure 11. Chart A-4: Shut down CICS

Notes for chart A-4

1. Notify user of impending shutdown

Broadcast a shutdown message to users about 20 minutes before planned shutdown to enable them to complete their work in an orderly manner.

2. Wait for users to finish

If a user is busy with a conversational transaction, the shutdown message is not received until the conversation is finished. CICS does not terminate while tasks are still active, so you must wait for tasks to finish to complete a normal shutdown.

3. Perform normal shutdown

When you are ready to shutdown CICS, enter the master terminal shutdown command:

```
CEMT PERFORM SHUTDOWN
```

When this shutdown command is entered, CICS displays the following messages at the console:

```
DFHTM1715 applid CICS/ESA is being quiesced by userid userid in transaction
          tranid[ at netname netname | at terminal terminal.]
DFHDM0102I applid CICS is quiescing.
```

This normal shutdown prevents CICS terminal users from starting any more transactions. If they attempt to do so, CICS ignores the terminal input and displays the message:

```
DFHAC2007 date time applid Transaction tranid cannot run as CICS
          shutdown is in progress
```

However, during the early part of the shutdown process, you can still enter the CICS master terminal transaction (CEMT); for example, to display system status. But when shutdown processing reaches a certain point, CICS stops all terminal activity (including CEMT) and waits for existing transactions to halt.

4. Successful shutdown

If normal shutdown is successful, CICS displays the following messages:

```
DFHZC2305I applid Termination of VTAM sessions beginning
DFHZC2316I applid VTAM ACB is closed
DFHJC4553I date time applid Archive job submitted for 01 A      1
DFHJC4510I applid All open journals now closed
DFHWK0105I applid Warm keypoint successful.
DFHXG6404I applid  SIGNING OFF NORMALLY FROM THE CAVM.        2
DFHXG6405I applid  SIGN OFF NORMAL FROM THE CAVM ACCEPTED.   2
DFHDU0303I applid Transaction Dump Data set DFHDMPB closed.
```

Notes:

1 You see this message only if you are using the CICS automatic journal archiving option. In this instance, the A data set of the system log was current at the time CICS was shut down, and an archive job is submitted automatically for DFHJ01A.

2 You see the CAVM sign-off messages only when you are running CICS with XRF=YES. In this instance, the shutdown is normal, and so is the sign off from the CAVM, which means that the CICS alternate is not called upon to perform a takeover.

If the shutdown messages do not appear in approximately 10 minutes of message DFHDM0102I (CICS is quiescing) being displayed, the system operator should notify the CICS master terminal operator, because there may be a problem with shutdown. In rare cases, the CICS region may still be in the first part of shutdown, during which the master terminal operator can enter CEMT transactions. If so, the status of any CICS tasks that are still running can be shown and, if necessary, they can be purged. This may permit normal shutdown to proceed.

Generally, if shutdown halts, it is too late for the master terminal operator to enter CEMT. In this case the CICS system must be terminated by the system operator.

When the warm keypoint is taken and journal data sets are successfully closed, the next start of CICS with START=AUTO results in a warm start.

5. Termination message

After message DFHDU0303I, the CICS job step ends with the usual MVS job step termination message:

```
JOB 'jobid'      'jobname'  'stepname'   DFHSIP      0000
```

If, when you have successfully shut CICS down, there are no more data-sharing subsystems running, you can also stop the IRLM. To do this, use the MVS command STOP *irlmproc*. This results in the following messages to the system console:

```
DXR032I IRLM STOP COMMAND ACCEPTED
DXR011I IRLM END-OF-TASK CLEAN-UP SUCCESSFUL
JOB jobno      IRLMJOB          DXRRLM00      0000
```

6. Cancel CICS

The MVS CANCEL command must be used. If this happens, an emergency restart is required to start up again, which CICS performs automatically with START=AUTO coded as the start parameter.

7. Copy disk logs

If you are using disk logging and DBRC, ensure that your system-log data sets (DFHJ01A and DFHJ01B) are copied using the IMS log archive utility, DFSUARC0. If you are using the automatic journal archiving option, you should ensure that the copy step in the skeletal JCL of the archive job uses DFSUARC0.

8. Complete log sheet

Complete the CICS online log sheet with stop date, time, and your name. Enter complete and accurate label details. If the shutdown is not normal, note this in the "Comments/incidents" area. This CICS online log sheet is used in the following CICS startup.

9. Complete the incident report

Complete the incident report and pass to the DC Administrator for follow-up.

Chart B-1: Action on user problem

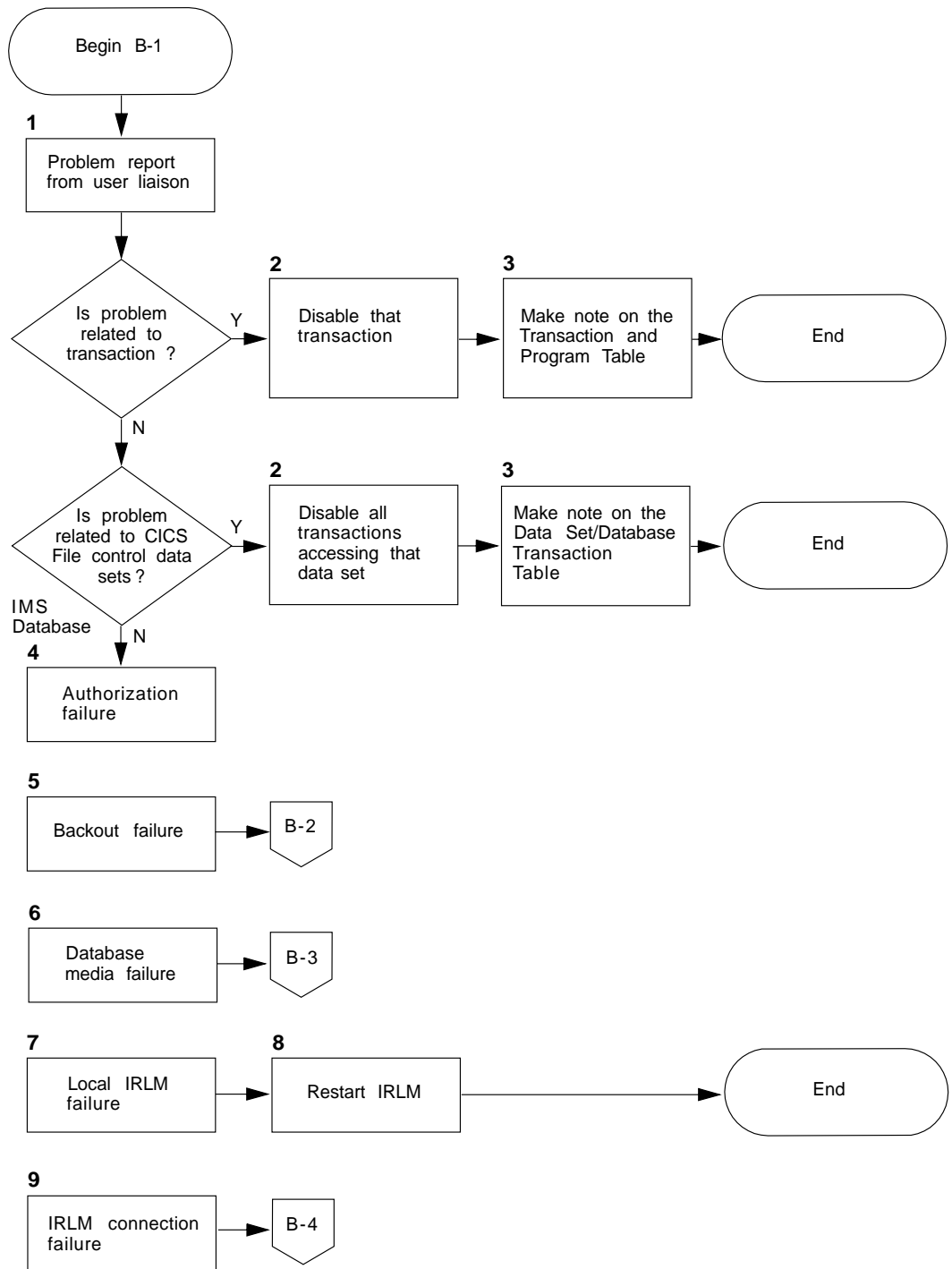


Figure 12. Chart B-1: Action on user problem

Notes for chart B-1

1. Problem report from user liaison

A remote terminal operator or end user who has a problem should contact user liaison. Your user liaison people determine the cause of the problem and decide on the action to be taken. If the problem is related to the transactions, or to the data sets or databases, your user liaison people may ask the CICS master terminal operator to disable one or more transactions, by submitting a formal request. This request could be on the "Transaction and program table" form for a transaction error, or on the "Data set and database and transaction table" form for a data set or database error.

2. Disable the transaction

Disable the transactions according to the request instruction by entering CEMT SET TRANSACTION(xxxx,yyyy,....) DISABLED.

Alternatively, depending on the complexity of the transactions that access the various databases, you might prefer to stop access to the databases.

3. Make note of the disabled transactions

After you have disabled the transactions, do not forget to complete the request by filling in the date, time, and your name on the form submitted to you.

4. Authorization failure

Authorization may fail with one of these messages:

```
DFS046A PROCESSING INTENT pppppppp FOR DATA BASE xxxxxxxx NOT COMPATIBLE
                                     WITH ACCESS yy. PSB=psbname.
DFS047A UNABLE TO OBTAIN AUTHORIZATION FOR DATA BASE xxxxxxxx. REASON
                                     CODE=zz. PSB=psbname.
```

Message DFS046A means that the access intent specified in the PSB for the database is more stringent than that defined in the ACCESS operand of the DFHDLDBD macro for the database. You must correct this incompatibility before you can process the database with this PSB. For further information about the meaning of the reason code in the DFS047A message, see the *IMS Messages and Codes manual*. Flowcharts B-2 through B-5 show the principal types of failure.

5. Backout failure

If a backout failure occurs, at least one DFHDL3905 (dynamic transaction backout) or DFHDL3906 (emergency restart) message appears. At least one DFHDL3907 message (see chart B-2) also appears. If the failure is caused by an I/O error, the following IMS message appears before the DFHDL390x messages.

```
DFS983I I/O ERROR IN DATA BASE xxxxxxxx , PROGRAM pppppppp DURING BACKOUT
```

In this case, forward recovery is also necessary, and chart B-2 indicates the required action.

6. Database medium failure

During CICS operation, a medium failure on an IMS database is indicated by a form of the message DFS0451, identifying the database, the program that detected the error, and the type of error.

```
DFS0451A|I modname,dbdname,ddname,i/o error,decb field
```

If the A-suffixed message is issued, the operator must respond. If the I-suffixed message is issued, processing continues. For information about the DFS0451 messages and their interpretation, see the *IMS Messages and Codes manual*.

7. Local IRLM failure

A local IRLM failure is indicated by a DXR message identifying the cause of failure. For information about these DXR messages, see the *IMS Messages and Codes manual*. Here are some of the possible messages:

```
DXR018E irlm INITIALIZATION ABEND
DXR019E irlm STORAGE POOL DESTROYED
DXR021E irlm ABEND VTAM ERROR
DXR022E irlm ABNORMALLY TERMINATED DUE TO SUBTASK FAILURE
DXR023E irlm INTERNAL OR MVS ERROR
```

If you suspect a local IRLM failure but have not received a message, display the list of active MVS jobs. If the IRLM is not active, it has failed.

8. Restart the IRLM

Use the normal batch job to restart the IRLM, then the CEMT PERFORM RECONNECT command to re-establish the connection between CICS and the IRLM. To reestablish communication between the local IRLM and an IRLM in another host, issue the MODIFY *irlmproc* ACTCOMM command to the nonfailing IRLM.

9. IRLM connection failure

One of the following messages notifies a failure of the connection between two IRLMs.

```
DXR025I irlm SESSION LOST, SHARING STATE IS zzzz
DXR027A irlm SESSION LOST, SHARING STATE IS IN DOUBT ACTION REQUIRED
```

Chart B-4 explains how to determine the cause of the failure, and tells you what action to take.

Chart B-2: IMS database backout failure

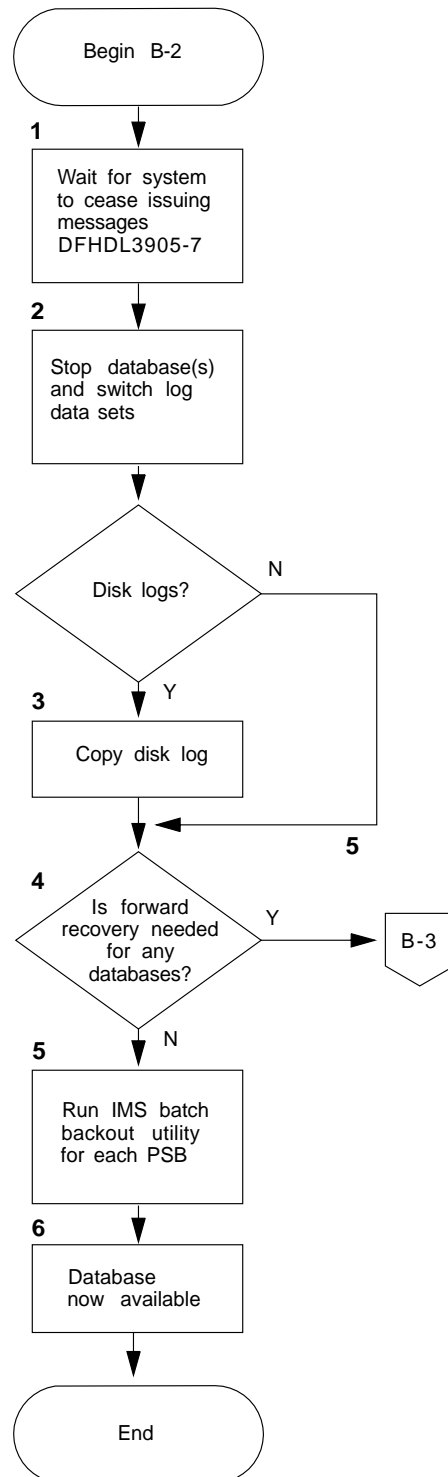


Figure 13. Chart B-2: IMS database backout failure

Notes for chart B-2

1. Wait for system to cease issuing messages DFHDL3905-7

The following messages are issued as a result of backout failure (DFHDL3905 in dynamic transaction backout, or DFHDL3906 in emergency restart):

```
DFHDL3905 date time applid Backout failure for PSB psb, transaction tranid.  
          Databases with uncommitted updates will be stopped  
DFHDL3906 date time applid Backout failure for PSB psb. Databases with  
          uncommitted updates will be stopped  
DFHDL3907 date time applid Database dbdname has been stopped
```

When a backout failure occurs, message DFHDL3905 or message DFHDL3906 is issued for each affected program specification block (PSB). Access to all affected databases is stopped, and message DFHDL3907 is issued for each database.

If the failure happens during dynamic transaction backout, all transactions scheduled with intent on those databases are abended with abend code ADLA on their next call to DL/I.

2. Stop databases

When you are satisfied that a message DFHDL3907 has been received for each database affected by the backout failure (for which a message DFHDL3905 or DFHDL3906 has been received), issue the CEMT SET DLIDATABASE(fileid) RECOVERDB command against all affected databases simultaneously, by including all the databases in one RECOVERDB command. This avoids switching the log more than once; each RECOVERDB command that you enter causes a log switch.

This has the following effects:

- New requests for access to the affected databases are rejected.
- Existing activity is allowed to quiesce; each affected transaction abends as explained in Note 1 above.
- The affected databases are stopped and deauthorized.
- A system log switch is forced (this allows the recovery job and CICS to run concurrently, each using a different log data set).

3. Copy disk log

Use the IMS log archive utility, DFSUARC0, to copy the now inactive log data set. If you are using the automatic journal archiving option, you should ensure that the copy step of the automatic archiving job uses DFSUARC0. This copy of the log is available for input to forward recovery and the IMS batch backout utility as required.

4. Is forward recovery needed?

If the backout failure has been caused by, or is combined with, a hardware problem on a database volume, then you must perform forward recovery for the affected databases before backout recovery. See chart B-3 on page 60 for details.

5. Run IMS batch backout utility

When running the IMS batch backout utility, you must:

- For dynamic backout failure, submit the log data set that was active when the failure occurred.
- For emergency restart backout failure, submit the log data set active at the abnormal CICS termination prior to the emergency restart.
- Run the backout utility **once** for each PSB named in a DFHDL3905 or DFHDL3906 message. Even if a PSB is named in more than one message, you should run the backout utility once only for that PSB.
- **Never** re-present a log data set that has previously been used to back out the same PSB, because this backs out subsequent valid updates.

6. Database now available

When recovery has been completed and databases are available for use, enter the CEMT SET DLIDATABASE(fileid) STARTED GLOBAL command once for each database. GLOBAL specifies that the SET STARTED command is to be transmitted to the data-sharing subsystems as well as being executed on the local CICS region.

Note: When backout failure occurs, DBRC records the failure against each database updated by the failing task. DBRC holds this information in the form of a “backout-needed” count in the database record in the RECON data set. This count is reduced by running the IMS database backout utility (see note 5). DBRC prevents access to the database until the count is reduced to zero.

Chart B-3: IMS database medium failure

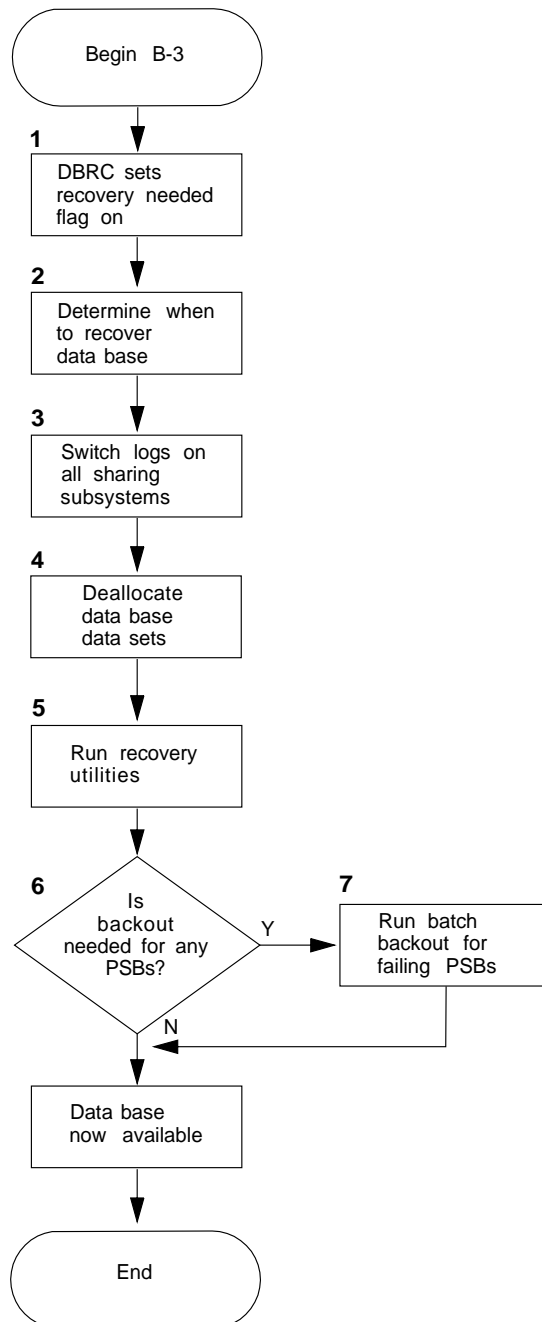


Figure 14. Chart B-3: IMS database medium failure

Notes for chart B-3

1. DBRC sets 'recovery needed' flag

Even though the 'recovery needed' flag is on, the database remains available to all authorized subsystems (unless the medium failure occurred during backout processing, in which case the database was stopped on all subsystems; see notes 1 and 2 for chart B-2 on page 57).

2. Determine when to recover databases

It is recommended that you allow batch jobs to complete (if possible) before you start recovery operations.

3. Switch log data sets on all sharing systems

The CEMT SET DLIDATABASE(fileid) RECOVERDB GLOBAL command, on one subsystem, ensures that all activity on the databases has completed. It also causes a log volume or disk data set switch in the issuing subsystem. A SET JOURNAL ADVANCE command should be issued in each other subsystem that holds authorizations to the database. As an alternative to the SET JOURNAL ADVANCE command, you can issue a local RECOVERDB to each subsystem.

Use the IMS log archive utility, DFSUARCO, to copy the now inactive log data set. If you are using the automatic journal archiving option, you should ensure that the copy step of the automatic archiving job uses the DFSUARCO utility.

4. Deallocate database data sets

If you need to perform forward recovery without interruption to CICS execution, you must first deallocate the database. You can use the CEMT SET DLIDATABASE(fileid) STOPPED command to do this.

Note: You must deallocate all data sets in the database. If this presents great difficulty, you may decide that it is preferable to shutdown CICS temporarily. Note that if a database has been allocated using IMS dynamic allocation, it is dynamically deallocated by a CEMT SET DLIDATABASE(fileid) STOPPED (or RECOVERDB) command.

5. Run recovery utilities

You can generate the JCL for the recovery jobs by issuing DBRC commands via the CICS CBRC master terminal transaction. For example:

```
      CBRC GENJCL 'CA GRPNAME(name) . . . optional parms'  
or    CBRC GENJCL 'DSLOG optional parms'  
then  CBRC GENJCL 'RECOV DBD(name) DDN(ddname) . . . optional parms'
```

Note: When you type DBRC commands using the CBRC transaction, you must follow the DBRC keyword by a single quote (not a period as shown in the DBRC command syntax). (The closing quotation mark, as shown in the examples above, is optional.) Thus the change accumulation modifier of the GENJCL command (GENJCL.CA) is entered as GENJCL 'CA . . . , and LIST.DB DBD(name) becomes LIST 'DB DBD(name). For information about the DBRC commands, see the *IMS Utilities Reference manual*.

The advantage of using GENJCL commands is that DBRC ensures automatically that the JCL refers to the correct logs and database copies.

Whether or not you use GENJCL, you must run the jobs indicated in the GENJCL commands shown above, that is run:

1. The IMS change accumulation utility, **or** the DBRC data set log utility.
2. The IMS database recovery utility.

6. Is backout needed for any PSBs?

Depending on the circumstances of the medium failure, backout may or may not be necessary. If the database that has been recovered was the subject of message DFHDL3907 (see chart B-2 on page 57), ensure that the processing described in chart B-2 is performed.

7. Run batch backout for failing PSBs

Run the IMS batch backout utility once for each failing PSB. Ensure that you provide as input every log current since the original I/O error occurred.

Chart B-4: IRLM connection failure

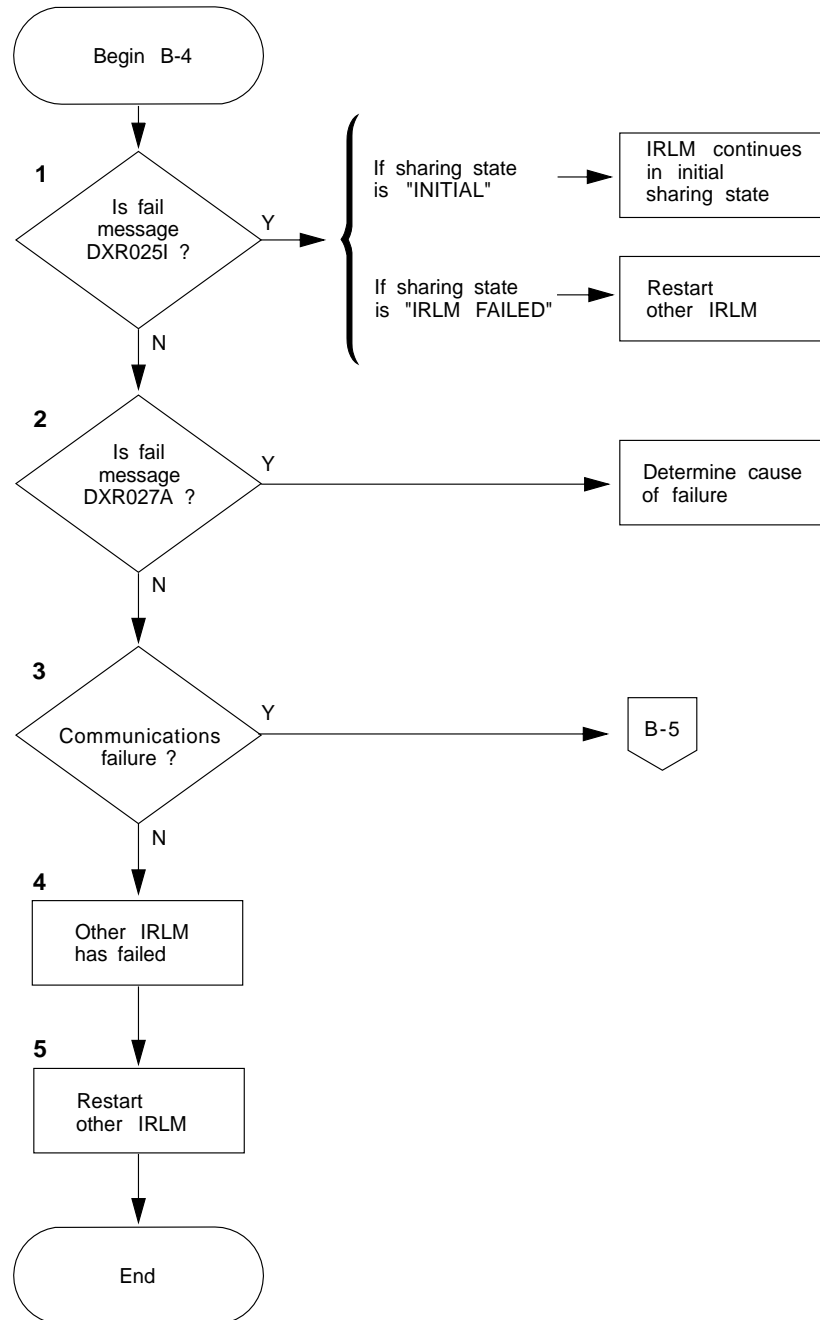


Figure 15. Chart B-4: IRLM connection failure

Notes for chart B-4

1. Message DXR025I, irlm SESSION LOST

The IRLM connection failure may produce this message:

```
DXR025I irlm SESSION LOST, SHARING STATE IS zzzz
```

Depending on the sharing state (zzzz), message DXR025I means either that the local IRLM cannot establish a session with the remote IRLM, that the remote IRLM has failed, or that communications have failed.

- If the sharing state is INITIAL, it means that the local IRLM has failed to establish a session with the remote IRLM. This normally is due to a VTAM problem. In the INITIAL state, the IRLM:
 - Is operational
 - Is not in communication with another IRLM
 - Has no retained information about another IRLM.

- If the sharing state is FAILED, it means that the remote IRLM has failed. All globally-shared databases that are authorized for update by subsystems using the failed IRLM are stopped and deauthorized, and all transactions scheduled to those databases are abended and the PSBs backed out. All DL/I transactions in subsystems using the failed IRLM are abended, and no PSBs referencing the stopped databases can be scheduled successfully. When the failing IRLM has been restarted, you must re-establish communications by issuing the following modify command to the **non-failing** IRLM:

```
MODIFY irlmproc, ACTCOMM
```

- If the sharing state is COMM FAILED, see notes 2 and 3.

2. Message DXR027A, irlm SESSION LOST

The IRLM connection failure may produce this message:

```
DXR027A irlm SESSION LOST, SHARING STATE IS INDOUBT ACTION REQUIRED
```

Message DXR027A indicates one of the following two things:

1. The remote IRLM failed and the local IRLM entered the in-doubt state.
2. Communications between the IRLMs failed and both entered the in-doubt state.

You must determine which of the above has occurred before you take any action. If the remote IRLM has also issued the DXR027A message, then it too is in doubt and a communications error has occurred. If the remote IRLM does not appear in a display of jobs active on its host system, then it has failed. ***If a communications failure coincides with a failure of the remote IRLM, you must treat it as a failure of the remote IRLM.***

3. Communications failure

Do **not** go to chart B-5 if the remote IRLM failed; see note 2.

4. Other IRLM failed

Issue the following MODIFY command for the in-doubt IRLM:

```
MODIFY irlmproc,SETSTATE,IRLM
```

The following messages can appear as a result of issuing the MODIFY command:

DXR014E Error in MODIFY command; required parameter omitted or inapplicable parameter included.

DXR034I A valid SETSTATE,IRLM has already been issued. Do not re-enter the command.

DXR026I Normal response to the specified command.

DFS630I PSBs are being backed out.

For information about these messages, see the *IMS Messages and Codes manual*.

Before you try to re-establish communication with the remote IRLM, make sure that the local IRLM has entered initial state by issuing the command:

```
MODIFY irlmproc,STATUS
```

When status is INITIAL, you can proceed to the next step.

5. Restart the other IRLM

When the failing IRLM has been restarted, you must re-establish communications by issuing a `MODIFY irlmproc,ACTCOMM` command to the IRLM that did not fail.

Chart B-5: IRLM communications failure

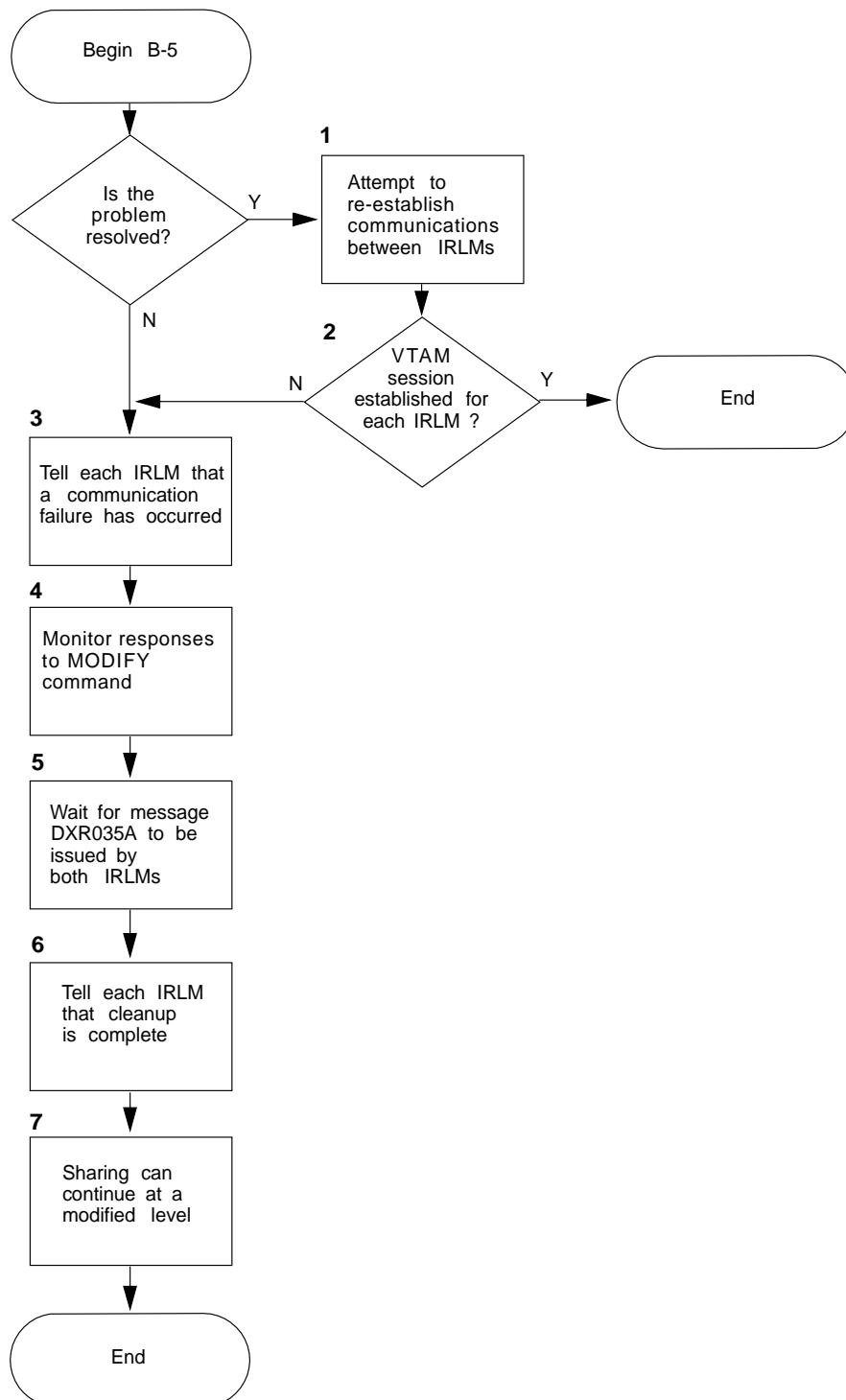


Figure 16. Chart B-5: IRLM communications failure

Notes for chart B-5

1. Attempt to re-establish communications

Each IRLM has issued either message DXR027A indicating that the sharing state is in doubt, or message DXR025I with "SHARING STATE IS COMM FAILED". It is possible that you can solve the problem quickly (for example, by issuing a VARY NET, ACTIVE command for a failing LU), or that it was transitory and has disappeared. If this is so, try to re-establish communication by entering to the local IRLM a `MODIFY irImproc,ACTCOMM` command.

If message DXR030 appears, the VTAM ACB is already open for the local IRLM, so issue the `F irImproc,ACTCOMM` command to the remote IRLM.

2. VTAM session established for each IRLM?

If you are successful in re-establishing communication between the IRLMs, each IRLM issues the message DXR002I:

```
DXR002I ir1m x VTAM SESSION WITH y ESTABLISHED
```

If either IRLM does not issue DXR002I, a communication problem still exists, and you should follow the charted procedure (point 3 and following).

3. Tell each IRLM of communication failure

You must issue the `MODIFY irImproc,SETSTATE,COMM` command to each IRLM, that did not issue message DXR025I with "SHARING STATE IS COMM FAILED".

The purpose of the `SETSTATE,COMM` command is to resolve the in-doubt state of the IRLM, by telling it that a communications failure has occurred. You must issue the command to each IRLM. As a result of the command, each subsystem using the IRLM stops any database authorized to a subsystem identified to the remote IRLM, abends, and backs out any transaction that has intent on such a database. DBRC deauthorizes all such databases to every local subsystem (message DFS2012 is issued—see note 4). Other authorizations are retained.

It follows from the above that, if one IRLM completes cleanup (that is, issues message DXR035A) before the `SETSTATE,COMM` command is issued to the second IRLM, then all subsystems of the second IRLM retain all their authorizations. This means that they can continue full processing without waiting for the restoration of the communications link.

You can take advantage of this in your planning for the situation where the sharing state of **both** IRLMs is in doubt (that is, they have both issued message DXR027A). Group your most critical transactions in one host. Then, in your operating instructions for a communications failure, include the following steps in the sequence shown:

1. Issue `SETSTATE,COMM` command to the non-critical IRLM.
2. Wait for message DXR035A.
3. Issue `SETSTATE,COMM` command to the critical IRLM.

4. Monitor responses to the MODIFY command

These messages can appear at this stage:

DXR014E irlm INVALID MODIFY COMMAND PARAMETERS

Error in MODIFY command; required parameter omitted or inapplicable parameter included. Correct command and re-enter.

DXR034I irlm zzzzzzzz REJECTED, xxxx-failed, CLEANUP NOT COMPLETE

A valid SETSTATE,COMM has already been issued. Do not re-enter command.

DFS2012 GLOBALLY SHARED DATA BASE dbname STOPPED

A subsystem found databases authorized to a subsystem of the remote IRLM. The stopping and deauthorization applies to all subsystems of *this* IRLM.

5. Wait for DXR035 messages

The cleanup process described in note 3 is initiated either by the SETSTATE,COMM command or automatically if message DXR025I was issued with "SHARING STATE IS COMM FAILED".

Message DXR035A is issued by IRLM on completion of the cleanup (that is, when all transactions using databases have completed or abended):

```
DXR035A irlm ENTER SETSTATE,INIT WHEN BOTH IRLMS ISSUE THIS MESSAGE
```

This message must be issued by each IRLM before the next step.

6. Tell each IRLM that cleanup is complete

Issue the MODIFY *irlmproc*,SETSTATE INIT to each IRLM. The SETSTATE,INIT command restores the IRLM to the INITIAL sharing state. When it has been issued, the IRLM is ready to participate in intrahost block-level data-sharing. It is also ready to resume communication with the remote IRLM.

You can now concentrate on solving the original problem, the failure of the VTAM link. When the link is restored, an ACTCOMM command to either IRLM restores communication and allows interhost block-level data-sharing to recommence.

7. Sharing can continue

The cleanup after communications failure may have left some databases authorized to subsystems in one host (see note 3). Such databases are now available for intrahost sharing in that host, and are unavailable to the other host. An unauthorized database can be authorized to a subsystem in either host, but is then unavailable to the other host.

Chart B-6: Machine check, power failure, MVS loop, or abend

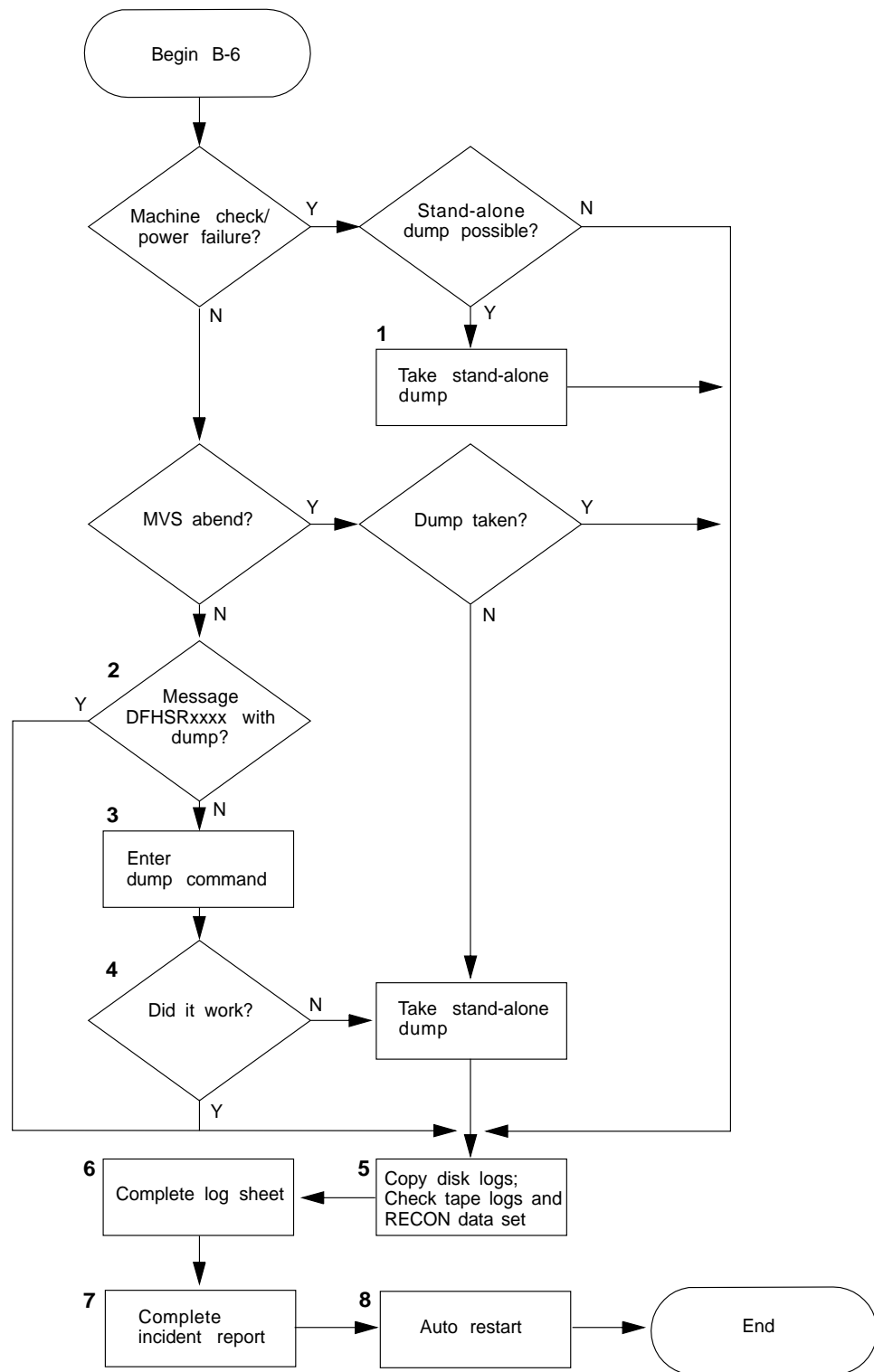


Figure 17. Chart B-6: Machine check, power failure, MVS loop, or abend

Notes for chart B-6

1. Take stand-alone dump

For a machine check or loop, a dump is useful to customer engineers and program service representatives to determine the cause of the problem. If possible, run an SADMP (stand-alone dump) program to obtain a stand-alone dump. The stand-alone dump program must be prepared on an active system before the failure occurs. For information about how to create and use a SADMP program, see the *MVS/ESA Service Aids* manual.

2. Message DFHSRxxxx with dump?

With the message DFHSRxxxx on the operating system console, you can check in the *CICS/ESA Messages and Codes* manual to see whether a system dump is provided (on a SYS1.DUMP data set) when CICS is abnormally terminated.

3. Enter dump command

If the failure causing the DFHSRxxxx message does not also produce a system dump, enter the MVS DUMP command through the console:

```
DUMP COMM=(dump-id).
```

MVS replies with the message:

```
nn IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND,
```

Reply to message IEE094D by entering the dump operands, for example:

```
R nn,JOBNAME=cicsrun
```

For information about the dump operands that you can use, see the *MVS/ESA System Messages Volume 1* and the *MVS/ESA System Messages Volume 2* manuals.

4. Did it work?

The operating system message IEA911E COMPLETE DUMP ON SYS1.DUMPnn tells you that the dump is successfully written in the SYS1.DUMPnn data set.

5. Copy disk logs

Copy the disk log data sets, and check tape logs and the RECON data set:

- For disk logs, you must save the contents before the emergency restart of CICS because CICS reuses the log data sets on restart. Use DFSUARC0 to save a disk log.
- If you have a tape log, and want to use it for offline recovery before restarting CICS, run the DFHTEOF utility to ensure that a tape mark has been written. This is not necessary if you are not performing any offline recovery, as the CICS emergency restart procedure runs DFHTEOF.
- Examine the RECON data set (DBRC recovery control data set) to ensure that the CICS stop time has been recorded in the PRILOG record. To do this, you can use the DBRC batch utility, DSPURX00, or the CBRC command on another CICS region, or the IMS DC DBRC online command facility. If it is necessary to update the PRILOG record, you can use one of the same three methods for a tape log. For a disk log, DFSUARC0 updated the PRILOG record when it copied the log.

Take care with the DBRC command to update the PRILOG record for tape logs. Use the following command:

```
NOTIFY.PRILOG RLDS RUNTIME(timestamp) STARTIME(timestamp) SSID(name)
```

You are advised not to issue the above command until you know that the tape log has been closed by either DFHTEOF or normal emergency restart processing.

- Due to the operating system or machine failure, CICS may have been unable to inform DBRC of the abnormal termination of CICS. Therefore DBRC may have a subsystem record for the failed CICS subsystem that indicates that no abnormal termination has occurred (SUBSYS record parameter ABTERM is set to OFF). It is possible to alter the contents of this subsystem record using the DBRC utility DSPURX00 with the CHANGE.SUBSYS statement, but this is not necessary as DBRC adjusts the RECON data set correctly when CICS restarts.

6. Complete the log sheet

Complete the CICS online log sheet with stop date, time, and your name. Make sure that details of tape copies of disk logs are correctly entered. State in the "Comments/incidents" area that the next startup must be an emergency restart. The information on this CICS online log sheet is used in the following CICS start-up.

7. Complete the incident report

The CICS incident report is used here for follow-up by DC Administration.

8. Restart CICS with START=AUTO

If a machine check or power failure occurs, you can perform an emergency restart (START=AUTO resulting in an emergency restart) as soon as the machine is up again, unless a disk failure is suspected.

Note: All failing subsystems must be restarted. No other subsystem can access segments for which IRLM holds locks for a failing subsystem until the failing subsystem has been restarted.

Errors that do not cause CICS to abend

This section describes some of the error conditions that can occur while CICS is running.

Backout failure during dynamic transaction backout or emergency restart

When a backout failure occurs, message DFHDL3905 (dynamic transaction backout) or DFHDL3906 (emergency restart) is issued for each affected PSB:

```
DFHDL3905 date time applid Backout failure for PSB psb, transaction tranid.  
          Databases with uncommitted updates will be stopped  
DFHDL3906 date time applid Backout failure for PSB psb. Databases with  
          uncommitted updates will be stopped
```

All affected databases are stopped, and message DFHDL3907 is issued for each database.

```
DFHDL3907 date time applid Database dbdname has been stopped
```

If the failure happens during dynamic transaction backout, all transactions scheduled with intent on those databases are abended with abend code ADLA on their next call to DL/I.

For information on how to recover from a backout failure, see chart B-2 on page 57.

Database stopped by another subsystem

Message DFHDL3908 is issued when a shared database is stopped as a result of a failure in another named subsystem.

```
DFHDL3908 date time applid Database dbdname has been stopped on request
          from subsystem sysid
```

For a CICS subsystem, the name used is the APPLID parameter specified in the CICS system initialization parameter.

Note: Any transactions currently scheduled to the database are abended with abend code ADLA. When a database is stopped, IMS prevents future tasks from accessing it.

When the database is again ready for use, you can make it available to CICS by issuing a local CEMT SET DLIDATABASE(fileid) STARTED command from this system, or a global CEMT SET DLIDATABASE (fileid) STARTED from another subsystem. The database is then opened when the first PSB referencing the database is scheduled.

Global command transaction error

Message DFHDL3918 tells you that the global command processor cannot be started:

```
DFHDL3918 applid Unable to attach CSGX, the global command processor.
```

This means that global commands from sharing subsystems cannot be processed.

CSGX and its associated programs and transactions are included in the CSD group DFHDLI, which is supplied in the pregenerated system and included in the IBM-defined group list, DFHLIST. Ensure that the DFHDLI group is included in the group list that you use to bring up CICS. (Specify GRPLIST=listname in the SIT or as a SIT override in the startup job stream.)

Failure to reconnect IRLM

When IRLM is restarted after a failure, you attempt to reconnect with a CEMT PERFORM RECONNECT command. If the command is rejected because of a failure to IDENTIFY (sign on to) the IRLM, message DFHDL3921 is issued, displaying the IMS IRLM IDENT request return code:

```
DFHDL3921 date time applid Identify failed during reconnect - RC = rr
```

For an explanation of the return code, see the *IMS Messages and Codes manual*. Then take the indicated action and retry the CEMT PERFORM RECONNECT command.

Global command transaction abend

Message DFHDL3925 tells you that the global command processor has abended.

```
DFHDL3925 date time applid Global command task CSGX has abended. No further
          commands received by this system can be processed.
```

This means that global commands from sharing subsystems cannot be processed.

The subsystem issuing the message can continue processing, and data integrity is maintained by DBRC. The operator of this system is at a disadvantage through not seeing global messages.

Abend errors

DBRC backout notification failure

Message DFHDL3911 is issued if DBRC is unable to process a request from CICS informing it that a backout failure has occurred.

CICS takes a system dump by issuing the SDUMP macro, and abends to preserve data integrity. In the dump, register 15 contains either a DBRC return code or an IMODULE return code. For the meaning of the return code, see the *IMS Messages and Codes manual*. Then take the necessary corrective action and restart CICS.

IRLM backout notification failure

Message DFHDL3913 is issued when IRLM has been unable to notify all sharing subsystems of a backout failure within CICS.

CICS takes a system dump by issuing the SDUMP macro, and abends to preserve data integrity. In the dump, register 15 contains the return code from the IMS macro, DFSLM. For the meaning of the return code, see the *IMS Messages and Codes manual*. Then take the necessary corrective action and restart CICS.

The IRLM failure processor cannot be started

Message DFHDL3919 is issued when CICS is unable to attach CSSX, the transaction that handles IRLM requests associated with failure conditions.

CICS takes a system dump by issuing the SDUMP macro, and abends to preserve data integrity.

CSSX and its associated programs and transactions are included in the CICS-supplied resource group DFHDLI. You should ensure that the DFHDLI group is included in the group list that you use to start up CICS (as specified on the GRPLIST=listname system initialization parameter).

IRLM request handler abend

Message DFHDL3920 is issued when CSSX (the transaction that handles IRLM requests associated with failure conditions) abends.

CICS takes a system dump by issuing the SDUMP macro, and abends to preserve data integrity.

Failure to reconnect to IRLM

Message DFHDL3922 indicates that IRLM has rejected a global command lock request, issued by CICS to effect a reconnect to IRLM.

CICS takes a system dump by issuing the SDUMP macro, and abends to preserve data integrity. In the dump, register 15 contains the IRLM return code.

For the meaning of the return code, see the *IMS Messages and Codes manual*. Then take the indicated corrective action and restart CICS.

System log inconsistent with DBRC

Message DFHDL3924 is issued during emergency restart if CICS finds a tas that requires DL/I backout but DBRC holds no authorizations for the affected databases.

CICS abends to preserve data integrity.

Check that the correct system log was being used. If not, perform an emergency restart with the correct log.

If the correct log was being used, you may have started a second CICS region with the same APPLID, which you must not do. Restart CICS with the correct APPLID.

Perform emergency restart

Start CICS with START=AUTO

When you specify START=AUTO, CICS examines the control record in the global catalog. If this indicates that the previous shutdown was abnormal, CICS performs automatically an emergency restart.

VSAM OPEN error messages

During the emergency restart, you may see a number of IEC161 messages, with return codes and function codes of 056-084 and 062-086. For example, the following messages are displayed when CICS opens the local catalog:

```
IEC161I 056-084,CICSNB,CICS,DFHLCD,,CICS410.applid.DFHLCD,  
IEC161I CICS410.applid.DFHLCD.DATA,QRS583.ICFCAT  
IEC161I 056-084,CICSNB,CICS,DFHLCD,,CICS410.applid.DFHLCD,  
IEC161I CICS410.applid.DFHLCD.INDEX,QRS583.ICFCAT  
IEC161I 062-086,CICSNB,CICS,DFHLCD,,CICS410.applid.DFHLCD,  
IEC161I CICS410.applid.DFHLCD.DATA,QRS583.ICFCAT
```

These indicate that the data set was not closed properly last time (056-084), and that OPEN routine has run the automatic VERIFY function successfully (062-086) to correct the previous abnormal termination. You see a set of these messages for each CICS system data set defined to VSAM.

Early in the initialization process you see the following message confirming that the startup is an emergency restart:

```
+DFHSI1502I applid CICS startup is Emergency.
```

Backout processing

Backout processing begins with these messages:

```
DFHRU2820I applid DFHRUP in progress  
:  
DFHRU2800I applid DFHRUP completed
```

The DFHRU2820I and DFHRU2800I DFHRUP messages indicate that the system-log data set in use at the time of the abnormal termination has been opened for input, and all the information needed for backout processing has been written to the restart data set, DFHRSD. The system log is then closed, ready to be opened for normal use later. For information about system log processing during emergency restart, see the *CICS/ESA Recovery and Restart Guide*.

The backout processing continues, with the various backout programs using the information written to the restart data set by DFHRUP. For example, if file control backout finds that no file needs to be backed out, you see the following messages:

```
+DFHFC5740I applid File backout beginning.  
+DFHFC5741I applid No file backout required.  
+DFHFC5742I applid File backout complete.
```


Copy disk logs

If you are using disk data sets for the system log, and specify JOURNAL=PAUSE, use DFSUARC0 to copy the emergency restart log when prompted by message DFHJC4583:

```
DFHJC4583 applid CICS SYSTEM LOG EMERGENCY DATA SET (DDNAME=DFHJ01X) READY  
TO BE COPIED REPLY 'Y01X' WHEN FINISHED
```

After a successful copy, run the DFHJCJFP utility to reformat the disk ready for reuse, and do not reply to the message until the format job has completed.

If you have specified the automatic journal archiving option for the system log (JOURNAL=AUTOARCH), ensure that the archiving step of the automatic archive job executes DFSUARC0. The DFHJC4583 message is not displayed when you are using automatic journal archiving; instead you see the DFHJC4553I message:

```
+DFHJC4553I date time applid Archive job submitted for 01 X 13:56:15
```

You do not need to reformat the DFHJ01X data set, or any other journal data set, after use if the automatic journal archiving option is used.

DBRC message

Message DFHDL3914 appears if database authorizations are held at the time of a previous system failure.

```
DFHDL3914 applid Retained DBRC authorizations for datasharing users have  
been released
```

IRLM message

Message DFHDL3916 appears only if one or more locks are held by IRLM for this CICS subsystem at the time of a previous system failure.

```
DFHDL3916 applid Retained IRLM locks have been released.
```

You receive this message if your CICS region supports MRO or the CICS shared database, and you have started the IRC session at system initialization:

```
+DFHSI1519I applid The interregion communication session was  
successfully started
```

Chart C-1: Log errors

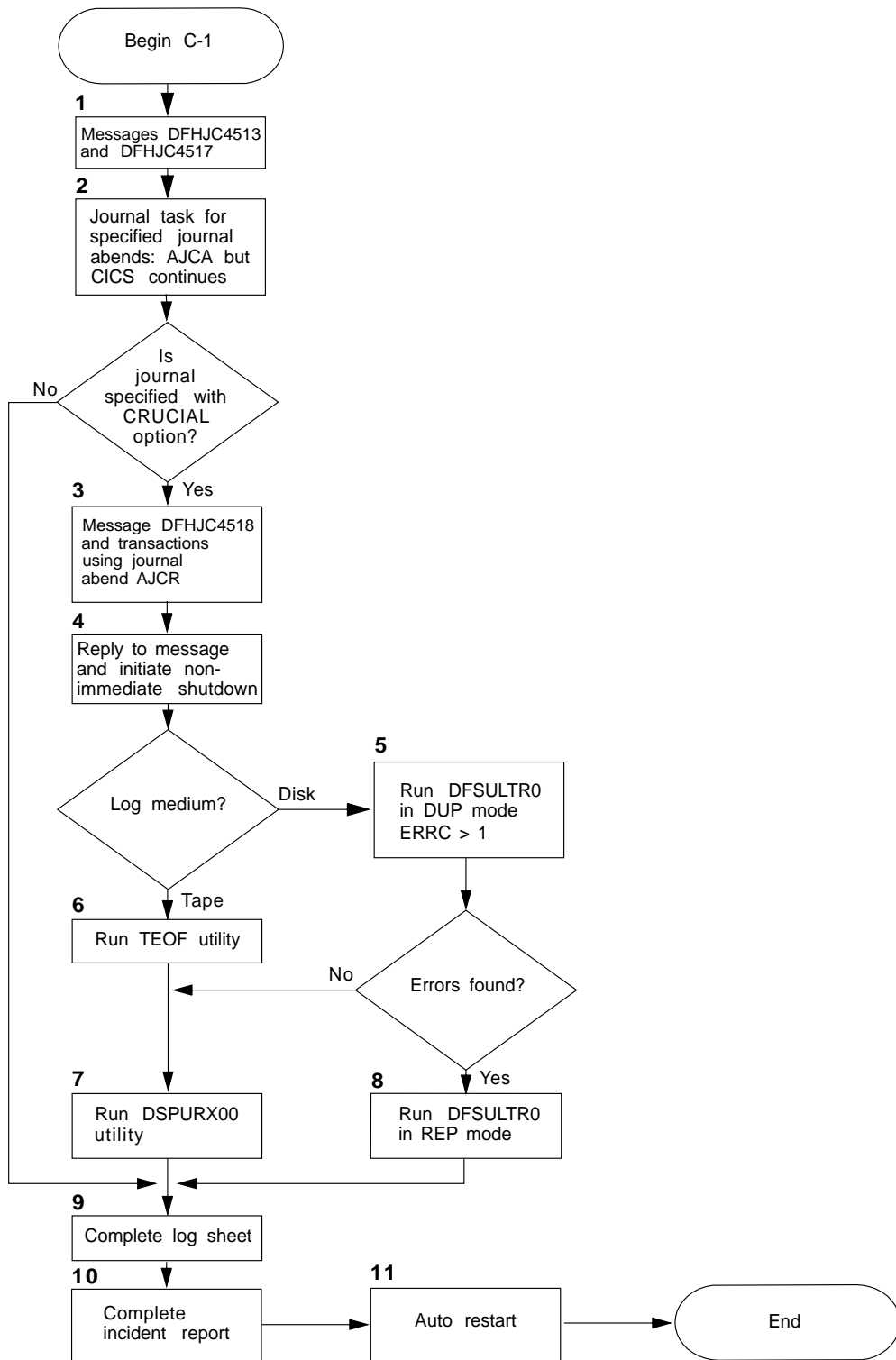


Figure 18. Chart C-1: Log errors

Notes for chart C-1

1. Messages DFHJC4513 and DFHJC4517

If an unrecoverable output I/O error occurs for a journal data set, CICS displays the message DFHJC4513 to indicate the specified journal, and DFHJC4517 to request that you initiate a non-immediate shutdown. In the following example, CICS issues message DFHJC4513 for the system log:

```
DFHJC4513 applid CICS system log no longer available - output I/O error
DFHJC4517 applid A non-immediate shutdown of CICS should be initiated
```

2. Journal task for specified journal abends AJCA

The journal task abends with abend code AJCA, but CICS continues executing.

3. Message DFHJC4518 and transaction abends AJCR

If the journal is specified with the CRUCIAL option in the DFHJCT TYPE=ENTRY macro instruction, CICS issues the message

```
DFHJC4518 applid Reply 'YES' to acknowledge message DFHJC4517
```

Transactions using the journal at the time of the failure abend with code AJCR.

4. Reply to message and initiate a normal shutdown

Because the journal is CRUCIAL, you should reply to the message as requested and initiate a non-immediate shutdown, using the CEMT PERFORM SHUT command.

5. Run DFSULTR0 in DUP mode

This step is an attempt to copy the disk log for forward recovery purposes. However, the original log at time of failure must be presented to emergency restart. Use DFSULTR0 as the copy utility, because the log was not closed properly as a result of the I/O error. For information about DFSULTR0, see the *IMS Utilities Reference manual*.

6. Run the CICS tape-end-of-file (DFHTEOF) utility.

DFHTEOF closes the data set in which the error occurred. Alternatively, you can run DFSULTR0 as for disk logs. DFHTEOF is also invoked automatically by CICS during emergency restart, to close the system log when it is defined on tape.

7. Run DSPURX00

Run this utility with a CHANGE.PRILOG statement as input to clear the log error flag in the RECON data set. For information about the command syntax, see the *IMS Utilities Reference manual*.

8. Run DFSULTR0 in REP mode

The purpose of running DFSULTR0 in REP mode is to recover the records in error, and to clear the log error flag in the RECON data set.

9. Complete the CICS log sheet

Complete the CICS online log sheet with stop date, time, and your name. For archived logs on tape, make sure that the allocated label column is correctly and completely filled in. State in the "Comments/incidents" area the reason for this CICS abend, and that the next startup must be emergency restart. The information on this CICS online log sheet is used in the following CICS startup.

10. Complete the incident report

The CICS incident report is used here for DC Administration follow-up.

11. Restart CICS

If, on the emergency restart, I/O errors occur when trying to read the log, emergency restart fails. If this happens, complete another incident report for the restart, and see DC Administration.

Chapter 5. Running a batch region for DL/I shared databases

The CICS-DL/I batch-region controller is a group of modules used instead of the IMS DB batch-region controller. The batch-region controller establishes and controls a dialogue between a CICS system, running with CICS local DL/I support, and the batch application program requesting DL/I access. The group of modules is supplied as part of the pregenerated CICS system, and is installed in the CICS libraries. The module DFHDRP is in CICS410.SDFHLOAD, and the modules DFHDRPA–G are in CICS410.SDFHAUTH.

CICS shared database support is provided by the interregion communications (IRC) component of CICS. To define the link between the MVS system and the batch systems that share the DL/I databases, define CONNECTION and SESSIONS resource definitions in the CSD, using the names @BCH and @BCH@B for the connection and session definitions respectively. For example:

```
CEDA DEFINE CONNECTION(@BCH) . .  
CEDA DEFINE SESSIONS(@BCH@B) . .
```

For guidance about how to code these definitions, see the *CICS/ESA Resource Definition Guide*.

You can set up the shared database session at system initialization with the system initialization parameter IRCSTRT=YES, or later by the CEMT SET IRC OPEN command.

CICS control of the IMS batch region needs certain parameters to be passed on the EXEC statement for the batch-region controller. Here is a sample job stream to set up a batch region:

```
//JOBNAME JOB   accounting info,name,MSGLEVEL=1           1  
//          EXEC PGM=DFHDRP,PARM=('SSA=xxxxx,PGM=xxxxxxxxx,', 2  
//          'PSB=xxxxxxxxx,CICS=xxxxxxxxx,LANG=x,CWTO=x','CMPAT=x')  
//STEPLIB DD   (User's own application programs)         3  
//DFHLIB DD   DSN=CICS410.SDFHAUTH,DISP=SHR              4  
//CHKDD DD   (Checkpoint data set(s))                   5  
//SYSUDUMP DD   SYSOUT=A  
//
```

Notes:

1 General points

Ensure that the latest level of module DFHC SVC, the CICS SVC, is installed in the link pack area, and defined to MVS as a Type 3 SVC.

The CICS region owns the DL/I databases and the journal, so do not include data definition statements for these files. Make sure that:

- All application control blocks (ACBs) needed by the sharing batch applications in the CICS region are entered in IMSVS.ACBLIB
- All program specification blocks (PSBs) are listed in the PSB directory (PDIR)
- All data management blocks (DMBs) are listed in the DMB directory (DDIR).

For information about DMBs and PSBs, see the *CICS/ESA System Definition Guide*.

2 You must specify most of the parameters in the list; for their possible values, see “Coding an EXEC PGM=DFHDRP statement” on page 81. You can use blanks or commas as separators, and blanks between keywords, equals signs, and option values. You can code the parameters in any order.

3 The STEPLIB libraries contain user-supplied application programs, and as such, should not be authorized. However, DFHDRP must be authorized, and therefore cannot be in any of the libraries defined in STEPLIB. It must be link-edited into an authorized library, such as SYS1.LINKLIB or another authorized library in the linklist. For an example of a link-edit job for DFHDRP, see the *CICS/ESA Installation Guide*.

4 The batch-region controller modules are installed in the CICS library CICS410.SDFHAUTH, which must be an authorized library. These modules are loaded from the CICS410.SDFHAUTH library (specified on the DFHLIB statement) when you run the DFHDRP program. If you give users access to the batch-region controller modules, you also give them access to the DFHSIP module, to which access should be strictly controlled. If you want to protect the other modules that are in the CICS410.SDFHAUTH library, such as the DFHSIP module, copy the batch region controller modules into another authorized library of your choice, and then amend the batch region job stream to specify the new library.

5 If the application contains any CHKP calls or commands that take an MVS checkpoint, specify at least one checkpoint data set in the job stream for the batch region. You can allocate one or two checkpoint data sets, if the batch-region controller provides checkpoint DCBs.

The associated ddnames for these checkpoint data sets must be CHKDD and CHKDD2 (if two data sets) or CHKDD (one data set). With two data sets, checkpoints are written alternately to each. The batch-region controller does not open the data sets before taking the MVS checkpoint. This means the third and subsequent checkpoints overwrite the previous ones so that both data sets only hold one checkpoint each at any one time. If only one data set is allocated, it is opened by the batch region controller; checkpoints are written consecutively and none is overwritten.

If checkpoint DCBs are provided by an assembler language application program, any number of checkpoint data sets may be used, with any ddnames. The application program may optionally open these data sets.

Coding an EXEC PGM=DFHDRP statement

The PARM field of the EXEC PGM=DFHDRP statement for the batch region controller may contain the following parameters (the first two are mandatory):

SSA=

The size of the largest segment search argument to be used by the batch application program in its database access. The value is in the range 9 to 32767 bytes. Specify it as accurately as possible; too large a value causes poor performance.

PGM=

The name of the application program; eight characters or less.

PSB=

The name of the program specification block. If no name is supplied, the application program name is the default.

CICS=

The name of the CICS system that contains the databases to be accessed. This should be eight characters or less. This is the name specified as the generic APPLID operand in the SIT or SIT override parameter. If you omit this operand, the default is DBDCCICS, which is also the default APPLID in the originating CICS system.

LANG={A|C|P}

Indicates the source language of the initial application program:

A for assembler
C for COBOL or C/370
P for PL/I

The default is assembler.

CWTO={Y|N}

Indicates whether, at a checkpoint request, the system issues a console message to tell the operator of a successful checkpoint. The default is Y.

CMPAT={N|Y}

Shows whether or not an I/O PCB is required by the application program for service calls. Code Y if the shared batch program is to use any system service calls such as CHKP.

Note: Code this CMPAT parameter with the value coded on the CMPAT parameter of the PSBGEN macro instruction used to generate the PSB. The default value for CMPAT on the PSBGEN macro is N; mismatched CMPAT operands cause the batch program to abend, with message DFH3717.

For information about program checks and abends occurring for IMS batch programs sharing DL/I databases, see the *CICS/ESA Problem Determination Guide*.

XRF considerations

Checkpoints are normally taken in batch jobs that do multiple units of work. After a takeover by the alternate, the alternate backs out any changes made following the last checkpoint. You need a checkpoint restart job to begin processing from the last checkpoint. You might prefer to make batch changes in short, separate, sequential jobs.

Chapter 6. Operating CICS in an XRF environment

This chapter describes the operation of a CICS region running with the system initialization parameter XRF=YES. It provides the following information about operating CICS in an XRF environment:

Page	Information
83	Outline of the benefits and use of CICS extended recovery facility (XRF).
85	Outline of the function provided for XRF by the CEMT and CEBT transactions.
88	Description of starting up and synchronizing active and alternate CICS regions in an XRF environment.
90	Description of shutting down active and alternate CICS regions normally and the main events that occur during normal shutdown.
94	Description of what events can cause an alternate CICS to takeover from an active CICS, what happens during a takeover, and any operator activity needed as a result of a takeover.
103	Description of the main points of a takeover that follows from a takeover command, and gives a flowchart for the sequence of these points.
106	Description of the main points of a takeover that follows when the active CICS signs off abnormally from the CAVM, and gives a flowchart for the sequence of these points.
108	Description of the main points of a takeover that follows from a missing surveillance signal, and gives a flowchart for the sequence of these points.
111	Description of the main points of a takeover that follows from the MVS image for the active CICS failing, and gives a flowchart for the sequence of these points. This form of takeover applies only to MVS/ESA SP 4.1 images.
111	Outline of what happens if the alternate CICS fails.
112	Description of how to use the program DFH\$AXRO, the CICS-supplied sample XRF overseer.

Overview of the CICS extended recovery facility (XRF)

The CICS extended recovery facility (XRF) provides a high level of availability to its end users by employing a degree of automated takeover from an active CICS region to an alternate CICS region, which can then effect a rapid emergency restart. The level of automatic takeover is outlined in this chapter.

If you are running CICS under MVS/ESA SP 4.1, CICS can use the cross systems coupling facility (XCF) of MVS to automate XRF takeover. This means that in the event of a MVS image failure, CICS XRF takeover to an alternate CICS on a different MVS image completes without the need for any operator intervention, if the following are all true:

- The active and alternate CICS are both running under MVS/ESA SP 4.1.

- Both MVS images are running in Processor Resource/Systems Manager (PR/SM) logical partitions (LPs), on the same central electronic complex (CEC) or on the same side of the CEC if it is physically partitioned.
- Both MVS images are defined to be in the same sysplex.
- You have defined to your XCF PR/SM policy² for dealing with MVS failures in a PR/SM environment (XCFPOLxx member of SYS1.PARMLIB).

Operator intervention is still required if, in the event of the MVS image for the active CICS region failing:

- The active and alternate CICS are running on separate MVS/ESA SP 4.1 images, in the same sysplex, but the MVS images are on different CECs or different sides of a physically-partitioned CEC. In this case, XCF asks the operator to confirm when a MVS image has terminated.
- The active and alternate CICS are running on separate MVS/ESA SP 4.1 images that are not in the same sysplex. In this case, CICS asks the operator to confirm when a MVS image has terminated.

If you are not running CICS under MVS/ESA SP 4.1, there are situations in which the alternate CICS cannot gather enough information to determine if it is safe to complete the takeover. For example, if you are running CICS in a multi-MVS³ XRF environment, and the MVS image on which the active CICS is running fails, the alternate CICS cannot receive reliable information about the state of the active CICS. In this situation, the operator must help the alternate CICS. You might also define some CICS regions in such a way that the operator must intervene to allow the alternate CICS to take over. This chapter describes these situations.

It is important that the operator has clearly-defined procedures to follow in as many problem situations as you can define for your installation. XRF is designed to give you a more rapid recovery capability, and you can only achieve that end if your operators can react correctly and quickly in those situations where a fully automatic takeover is not possible.

For information about running CICS with XRF, about XRF terminology, about the role of the operator, and of the responsibilities of the systems programmer towards the operator, see the *CICS/ESA 3.3 XRF Guide*.

For information about sysplex management, see the *MVS/ESA Planning: Sysplex Management manual*.

For information about running CICS-supplied installation verification procedures (IVPs) to test your CICS regions with XRF, see the *CICS/ESA Installation Guide*.

² XCF PR/SM policy: The function that enables MVS images to take over the resources of other MVS images in the same sysplex. This term is also known as the PR/SM automatic reconfiguration facility (ARF). For information about the XCF PR/SM policy, and its effect in single and multi-CEC environments, see the *MVS/ESA Implementation Guide*.

³ Multi-MVS environment: An environment that supports more than one MVS image. For example, this could be one physical processing system (such as an IBM 3090) that is partitioned into one or more processors, where each partition can run under the control of a single MVS operating system. Alternatively, if you are running MVS with the Processor Resource/Systems Manager (PR/SM), this could be multiple logical partitions, with each logical partition operating a copy of MVS. For further guidance information about all of the multi-MVS environments that support CICS running with XRF, and about XRF terminology, see the *CICS/ESA 3.3 XRF Guide*.

Functions of the CEMT and CEBT transactions for XRF

This section tells you about the extra function provided for XRF in CEMT, and also about the CEBT transaction. For information about using the CEMT and CEBT commands, see the *CICS/ESA CICS-Supplied Transactions* manual.

Master terminal transaction (CEMT)

CEMT is used to control the active CICS region, but not the alternate CICS, which is controlled by CEBT. For a summary of the effects of a CEMT command on CICS running with XRF=YES, see Tables 4 and 5 on page 86.

The command syntax for shutdown is:

```
CEMT PERFORM SHUTDOWN [IMMEDIATE|TAKEOVER] [DUMP]
```

When the CEMT PERFORM SHUTDOWN command is issued without any options, both the active CICS and the alternate CICS (if any) that was tracking it, shut down normally.

When the CEMT PERFORM SHUTDOWN command is issued with the TAKEOVER or IMMEDIATE option, the alternate CICS tries to take over instead of terminating, **unless it is not running or its takeover option is defined as COMMAND.**

Effect of the IMMEDIATE shutdown option: If you specify the IMMEDIATE option on the shutdown command, CICS terminates all active tasks and SNA sessions, and may cause CICS to abend if tasks are still running.

Effect of the TAKEOVER shutdown option: If you specify the TAKEOVER option on the shutdown command, CICS performs a normal shutdown before signing off abnormally from the CICS availability manager (CAVM). Only then does the alternate CICS (if any) attempt the takeover. This means that the takeover option is, in general, slower than the immediate option, and may be more suited to those occasions when you want to do some kind of planned maintenance.

Effect of the TAKEOVER option with no alternate CICS running: If you intend to use the CEMT PERFORM SHUTDOWN TAKEOVER command on an active CICS running with XRF=YES, first ensure that the alternate CICS is running. If you enter this shutdown command when the alternate CICS is not running (for example, if it has not started up), the active CICS shuts down, but does not issue any messages telling you that the alternate CICS cannot take over.

Effect of the alternate CICS takeover option TAKEOVR=COMMAND: If TAKEOVR=COMMAND is specified for the alternate CICS, the alternate CICS remains in standby mode, even after the active CICS has signed off abnormally, until you issue a specific CEBT PERFORM TAKEOVER command through the console of the alternate CICS.

The following tables summarize the effects of the CEMT command issued to the active CICS region:

<i>Table 4. CEMT PERFORM SHUTDOWN, effect on active CICS</i>			
Activity	Normal	Takeover Option	Immediate Option
Tasks	Finish	Finish	Terminate
SNA-session XRF	Unbind	Leave	Unbind
SNA-session non-XRF	Unbind	Unbind	Unbind

<i>Table 5. CEMT PERFORM SHUTDOWN, effect on alternate CICS</i>			
Activity	Normal	Takeover Option	Immediate Option
alternate CICS with TAKEOVR=AUTO or MANUAL	Shuts down	Takes over	Takes over
alternate CICS with TAKEOVR=COMMAND	Shuts down	Remains in standby mode	Remains in standby mode

Master terminal transaction (CEBT)

CEBT is used to control an alternate CICS before takeover occurs. The information presented here is a summary of the information about CEBT in the *CICS/ESA CICS-Supplied Transactions* manual. There is also some information about CEBT in the *CICS/ESA 3.3 XRF Guide*.

From an operating standpoint, CEBT differs from CEMT in that:

- The only way you can use it is by issuing the MVS MODIFY command, usually through the system console
- Its use is not limited by signon or security key.

Remember that CEBT runs in an environment in which CICS resources have not been recovered; that is, in the alternate CICS, before takeover has occurred and emergency restart has been completed.

You can use CEBT to control the following functions:

TAKEOVER

CEBT PERFORM TAKEOVER [DUMP] causes the alternate CICS to take over from the active CICS.

CEBT SET TAKEOVER AUTOMATIC|MANUAL|COMMAND enables you to change the option specified on the TAKEOVR system initialization parameter, while the alternate CICS is running.

CEBT INQUIRE TAKEOVER displays the setting of the takeover option.

SHUT DOWN

CEBT PERFORM SHUTDOWN [IMMEDIATE] [DUMP]

PERFORM SHUTDOWN causes the alternate CICS to shut down. All active system tasks and SNA standby sessions are terminated.

If IMMEDIATE is specified then the alternate CICS signs off abnormally from the CAVM. If not, it signs off normally from the CAVM. There are no other differences, given the environment in which CEBT is used.

If DUMP is specified then a system dump is produced on completion of the termination process.

SURVEILLANCE

CEBT SET SURVEILLANCE ON allows the alternate CICS to attempt a takeover when the active CICS surveillance signal is overdue, subject to the current setting of the takeover option.

CEBT SET SURVEILLANCE OFF inhibits the alternate CICS from attempting a takeover when the active CICS surveillance signal is overdue.

CEBT INQUIRE SURVEILLANCE displays the surveillance as ON or OFF.

DUMP

CEBT INQUIRE DUMP

CEBT SET DUMP
[OPEN|CLOSED|SWITCH]
[AUTOSWITCH|NOAUTOSWITCH]

These commands provide the same function for managing the CICS transaction dump data sets of an alternate CICS as CEMT does for an active CICS.

CEBT PERFORM SNAP causes CICS to issue the MVS SDUMP macro to produce an SDUMP of the alternate CICS.

INTERNAL TRACE

CEBT INQUIRE INTTRACE

CEBT SET INTTRACE
[START|STOP]

These commands have only a limited function for managing an internal trace for an alternate CICS, compared with the CETR command for an active CICS.

AUXILIARY TRACE

CEBT INQUIRE AUXTRACE

CEBT SET AUXTRACE
[CUR]
[START|PAUSE|STOP]
[NOSWITCH|NEXT|ALL]
[SWITCH]

These commands have only a limited function for managing an auxiliary trace for an alternate CICS, compared with the CETR command for an active CICS.

GTF TRACE

CEBT INQUIRE GTFTRACE

CEBT SET GTFTRACE
[START|STOP]

These commands have only a limited function for managing a GTF trace for an alternate CICS, compared with CETR for an active CICS.

Starting up and synchronizing an XRF system

Essentially, the active CICS region is started in the same way as a CICS region with XRF=NO. (An example job log from starting a CICS region is given in Figure 3 on page 22.)

Synchronizing time-of-day clocks in a multi-MVS environment

In a multi-MVS environment, it is important to synchronize the time-of-day clocks as closely as you can (to within as few seconds as possible). If the clock of the active CICS is ahead of the clock of the alternate CICS, there is a delay during takeover, until the later clock is synchronized.

There is no comparable problem if the clock of the active CICS is behind that of the alternate CICS; it simply means that, in the event of a takeover, there is a gap (apparent time loss) in the time stamping of records written to data sets (such as system management facility (SMF) and CICS region log data sets).

In a multi-MVS/ESA SP 4.1 environment, the clocks of all MVS systems in the same sysplex must be synchronized. In a PR/SM environment, clock synchronization is guaranteed, because all MVS images share the same hardware clock. However, in a non-PR/SM environment all MVS images must share a sysplex timer. Either way, CICS assumes that the clocks of all MVS images in a sysplex are synchronized.

For example job logs from starting XRF CICS regions, and from an alternate XRF CICS region taking over from an active XRF CICS region, see *CICS/ESA Installation Guide*.

Starting up an active CICS

This section lists the main points in the startup of an active CICS region.

As a started task or MVS job

You can start an active CICS as a started task or submit an MVS job. If you use the MVS START command, you should write the procedure using symbolic parameters to qualify the data set names of those data sets that are unique to either the active or the alternate CICS. For example, the active CICS region has a unique local catalog (unlike the global catalog, which is shared). For information about CICS region data sets and XRF considerations, see the *CICS/ESA System Definition Guide*.

The job control statements for starting a CICS region with XRF=YES are similar to those for XRF=NO, with the addition of the DD statements for the CAVM data sets. You also require the XRF-specific system initialization parameters, specified either in the SIT, the PARM statement, or the SYSIN data set.

When running a production CICS region with XRF=NO, you normally specify all the system initialization parameters in the SIT. However, when you run CICS with XRF and, as recommended, you start the active and alternate CICS regions using the same SIT, you must specify some system initialization parameters in the PARM parameter or SYSIN. For example, the active and the alternate CICS require different START parameters, and also different specific APPLIDs. For example:

Active CICS – START=AUTO
 – APPLID=(DBDCCICS,DBDCCIC1)

Alternate CICS – START=STANDBY
 – APPLID=(DBDCCICS,DBDCCIC2)

For information about all the system initialization parameters, see the *CICS/ESA System Definition Guide*.

The active signs on to the CAVM

CICS signs on to the CAVM as the active CICS, and CAVM replies when the signon is accepted, with the following messages:

```
DFHXG6400I applid Signing on to the CAVM as active with generic APPLID genericid  
DFHXG6401I applid Sign on to the CAVM as active accepted
```

Note: Although an active and alternate CICS region each has its own CAVM, they are logically one, each communicating through the same pair of CAVM data sets.

Normal initialization continues

CICS continues normal initialization, after determining the type of startup from the START= parameter and the control record in the global catalog. If you specify the START=AUTO system initialization parameter, and the control record in the catalog shows that CICS previously shut down normally with a successful warm keypoint, the active performs a warm start.

Notifies VTAM of specific APPLID for this CICS

The active informs VTAM of the specific APPLID that VTAM should now associate with the CICS generic APPLID. It does this by issuing the MVS “MODIFY netname,USERVAR” command. For further information about this command, and the VTAM user variable that it sets, see the *CICS/ESA 3.3 XRF Guide*.

The active CICS detects the alternate CICS signing on to the CAVM

The alternate CICS signs on to the CICS availability manager, is detected by the active CICS, and acknowledged by message DFHXG6403I.

The active starts the catch-up phase

When initialization is completed, the active CICS enters the catch-up phase, passing information to the alternate CICS by way of the CAVM message data set.

Starting up an alternate CICS

This section lists the main points in the startup of an alternate CICS region.

As a started task or MVS job

You can start an alternate CICS as a started task or submit an MVS job. If you use the MVS START command, you should write the procedure using symbolic parameters to qualify the data set names of those data sets that are unique to either the active or the alternate CICS. For example, the active CICS region has a unique local catalog (unlike the global catalog, which is shared). For information about CICS region data sets and XRF considerations, see the *CICS/ESA System Definition Guide*.

It does not matter if you start the alternate CICS before its associated active CICS, although normally you ensure that the active CICS is successfully started before submitting the job for the alternate CICS.

The alternate CICS signs on to the CAVM

The alternate CICS signs on to the CAVM and notifies you when the signon is successful with the following messages:

```
DFHXG6500I applid Signing on to the CAVM as alternate with generic APPLID gapplid
DFHXG6501I applid Sign on to the CAVM as alternate accepted
```

The alternate CICS detects the active CICS

Having signed on to the CAVM, the alternate CICS detects, by way of the CAVM data sets, that the active CICS is running. This is indicated on the console by the following message:

```
DFHXG6503I applid Sign on of applid to the CAVM as active detected.
```

Starts terminal control catch-up and tracking

When the initialization of the alternate CICS is suspended pending takeover, terminal control tracking and catch-up commences, with the following messages displayed at the console:

```
DFHTD0103I applid Transient Data initialization has been suspended pending takeover.
DFHTC1041I applid Terminal control tracking started.
DFHTC1044I applid Terminal control catch-up started.
      :
DFHTC1045I applid Terminal control catch-up complete.
```

The first stage of terminal control tracking is the catch-up phase, during which the active CICS notifies the alternate CICS, by way of the CAVM data sets, of the current state of all its VTAM terminals. For information about terminal control tracking, see the *CICS/ESA 3.3 XRF Guide*.

The alternate CICS continues surveillance and tracking

The alternate CICS is now in standby mode, but only partially initialized, and therefore cannot process normal transactions. In this standby mode, the alternate CICS performs the following tasks:

- Its primary function of monitoring the active CICS for any sign of failure
- Reading the CAVM messages of the active CICS, and using the information contained in them to maintain its own TCT, and establish standby sessions for XRF-capable terminals
- Processing any CEBT commands entered through the MVS console with the MVS/ESA MODIFY command.

Shutting down CICS XRF regions

If you shut down the active CICS normally, any alternate CICS that was tracking it also shuts down normally. If you do not shut down the active CICS normally, the alternate CICS takes over from the active CICS. Takeover is started either on explicit command, or because of actual or apparent failure of the active, subject to the setting of the TAKEOVR option in the SIT.

If you shut down the alternate CICS, the active CICS becomes aware of this because the alternate CICS signs off from CAVM. However, the active CICS continues running normally, but without any standby. You can start up another alternate CICS to accompany the active CICS whenever you require it.

Shutting down the active CICS normally

This section lists the main points in the shutdown of an active CICS region. (These points are also shown in the flowchart in Figure 19 on page 93.)

1. Issue the shutdown command

When you issue the shutdown command to the active CICS it performs normal CICS shutdown processing, beginning with the following messages:

```
DFHTM1715 applid CICS/ESA is being quiesced by userid userid in transaction
          tranid[at netname netname | at terminal terminal.]
DFHDM0102I applid CICS is quiescing.
DFHZC2305I applid Termination of VTAM sessions beginning
DFHZC2316I applid VTAM ACB is closed.
```

2. Sign-off from the CAVM

The active does not sign off from the CAVM until all tasks have terminated, and the warm keypoint has been successfully taken, as shown by the following sequence of console messages:

```
DFHJC4553 date time applid Archive job submitted for 01 B
DFHJC4510I applid All open journals now closed
DFHWK0105I applid Warm keypoint successful.
DFHXG6404I applid SIGNING OFF NORMALLY FROM THE CAVM.
```

For more information about a CICS shutdown, see chart A-4 on page 51.

3. Termination complete

To signify that termination is complete, CICS issues the following message to the console:

```
DFHKE1799 applid TERMINATION OF CICS/ESA IS COMPLETE
```

4. The sign-off is detected by the alternate CICS

When the active CICS signs off normally from the CAVM, the alternate CICS detects that the shutdown of the active CICS it was tracking is normal, and issues the following message to the console:

```
DFHXG6507I applid SIGN OFF NORMAL FROM THE CAVM DETECTED
```

5. The alternate CICS initiates its own shutdown

As a result of noting the normal shutdown of the active CICS, the alternate CICS starts its own shutdown, issuing the following message:

```
DFHXG6520I applid CICS shutdown initiated by CAVM event.
```

6. The alternate CICS signs off from the CAVM

After issuing the shutdown initiation message, the alternate CICS signs off from the CAVM with the following messages:

```
DFHXG6404I applid SIGNING OFF NORMALLY FROM THE CAVM
DFHXG6405I applid SIGN OFF NORMAL FROM THE CAVM ACCEPTED
```

7. Termination of the alternate CICS complete

To signify that termination is complete, CICS issues the following message to the console:

```
DFHKE1799 applid TERMINATION OF CICS/ESA IS COMPLETE
```

Shutting down the alternate CICS region normally

This section lists the main points in the shutdown of an alternate CICS region. (These points are also shown in the flowchart in Figure 20 on page 94.)

1. Issue the shutdown command

When you issue the CEBT PERFORM SHUTDOWN command to close an alternate CICS region, the alternate CICS starts to shut down, issuing the message:

```
DFHXG6521I applid CICS SHUTDOWN INITIATED BY CEBT EVENT
```

If you add the operand IMMEDIATE to the CEBT command, the active informs the operator with the message:

```
DFHXG6411I applid Sign off abnormal from the CAVM detected.
```

2. Sign-off from the CAVM

The alternate CICS signs off normally from the CAVM, issuing the following console messages:

```
DFHXG6404I applid SIGNING OFF NORMALLY FROM THE CAVM  
DFHXG6405I applid SIGN OFF NORMAL FROM THE CAVM ACCEPTED
```

3. Termination complete

The successful shut down of the alternate CICS is shown by the same MVS message as for the active CICS. There is no specific CICS message to indicate that the alternate CICS has terminated.

4. Active informs operator that the alternate CICS has signed off

When the alternate CICS signs off from the CAVM the sign-off is detected by the active CICS, which notifies the operator with the following message:

```
DFHXG6407I applid SIGN OFF NORMAL FROM THE CAVM DETECTED.
```

5. The active continues normal processing

The active CICS region continues normal processing, but without an alternate CICS as a standby system. However, the active CICS remains signed on to the CAVM, and you can start a new alternate CICS at any time. This is detected by the active, and surveillance and tracking resume after the catch-up phase is completed.

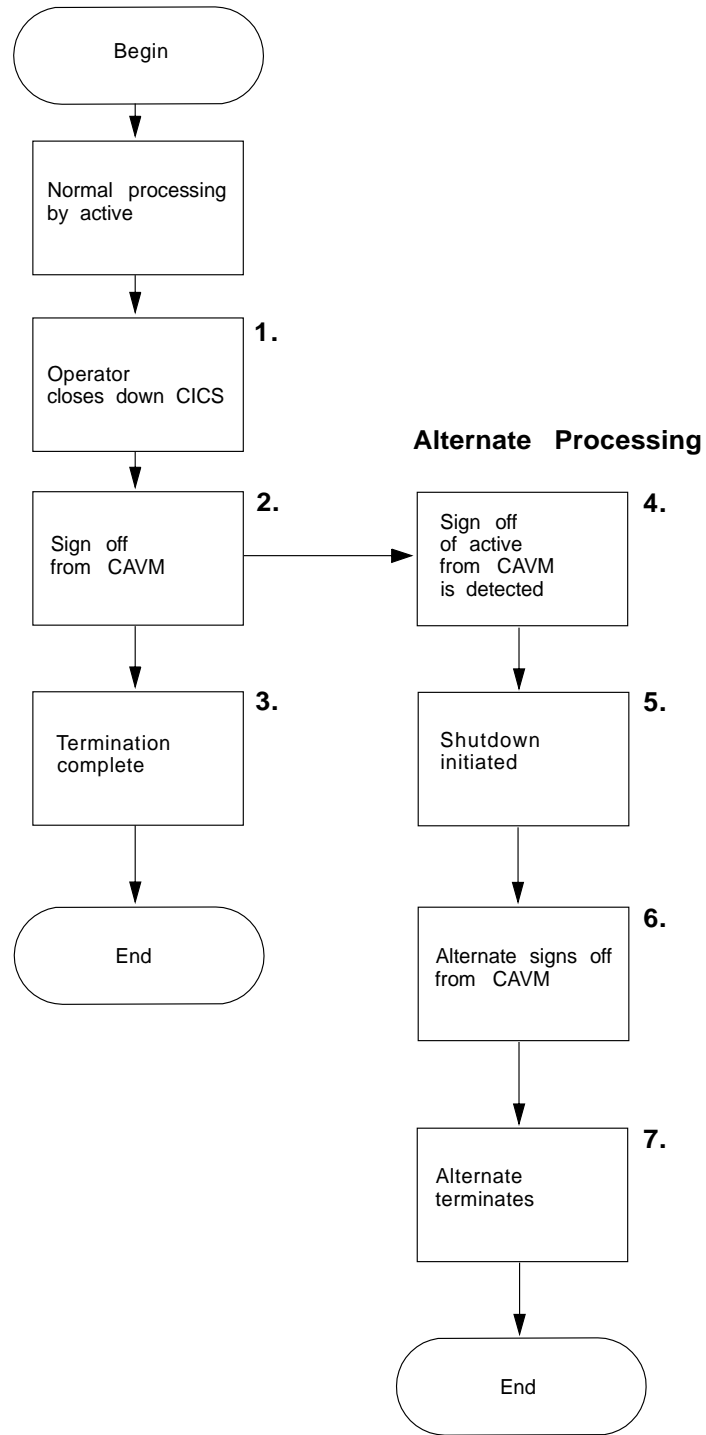


Figure 19. Shutting down the active CICS normally

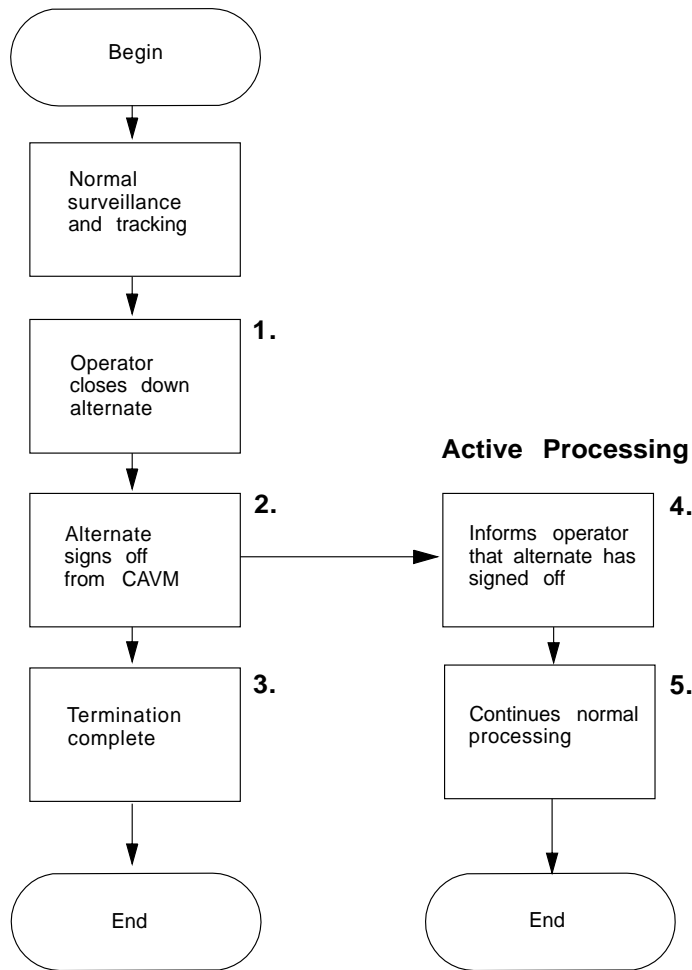


Figure 20. Shutting down the alternate CICS normally

Takeover

This section provides an overview of takeover activity. It discusses:

- What events can cause a takeover
- What happens during a takeover
- Post-takeover processing.

The alternate CICS monitors an active CICS, and takes no action as long as the surveillance signal is updated on time and no state change occurs. If any of the events outlined in “What events can cause a takeover?” on page 95 occur, the alternate CICS tries to takeover from its partner CICS.

What events can cause a takeover?

This section outlines the events that can cause an alternate CICS to take over from an active CICS. These events depend on the options specified in the TAKEOVR system initialization parameter of the alternate CICS.

- An operator at a console issues a command to initiate a takeover. This command can be either of the following:
 - CEBT PERFORM TAKEOVER, issued to an alternate CICS region. The CEBT PERFORM TAKEOVER command can also be issued by a master or coordinator region to a dependent region.
 - CEMT PERFORM SHUT TAKEOVER (or IMMEDIATE), issued to the active CICS region.

For more information, see “Takeover after a command” on page 103.

- An active system detects that it is failing and signs off abnormally from the CAVM. The alternate CICS is governed by the takeover options defined in the SIT, and acts as follows:
 1. If **TAKEOVR=AUTO** or **MANUAL**, the alternate CICS begins takeover processing.
 2. If **TAKEOVR=COMMAND**, the alternate CICS takes no action, and continues in standby mode, even though the job for the active CICS has terminated abnormally. It can only perform a takeover when it receives an explicit CEBT command through the console.

For more information, see “Takeover after abnormal sign-off” on page 106.

- An active CICS might fail, but not be able to sign off abnormally. The disappearance of the surveillance signal indicates possible failure to the alternate CICS, and it acts as follows:
 1. If **TAKEOVR=AUTO** or **MANUAL**, the alternate CICS begins takeover processing.
 2. If **TAKEOVR=COMMAND**, the alternate CICS takes no action, and continues in standby mode. Again, you have the choice of restarting the active CICS in place, or instructing the alternate CICS to takeover by means of an explicit CEBT command through the console.

For more information, see “Takeover after missing surveillance signal” on page 108.

- Under MVS/ESA SP 4.1 and PR/SM, the MVS image for the active CICS fails. Failure of the MVS image for the active CICS triggers an XRF takeover regardless of the value specified for the SIT parameter TAKEOVR in the alternate CICS, if the active and alternate CICS are running under different MVS/ESA SP 4.1 images in the same sysplex, in PR/SM LPs, on the same CEC, or on the same side of the CEC if it is physically partitioned.

If the active and alternate CICS are running under different MVS/ESA SP 4.1 images in the same sysplex, and the MVS images are running on different CECs, XCF asks the operator to verify that the MVS image has failed. When the operator verifies that the MVS image has failed, CICS detects this and automatically starts the CICS XRF takeover.

For more information, see “Takeover after MVS failure (MVS/ESA SP 4.1)” on page 111.

Figure 21 sets out the causes of a takeover event, and Table 6 on page 97 summarizes the takeover events and the types of takeover that they can lead to.

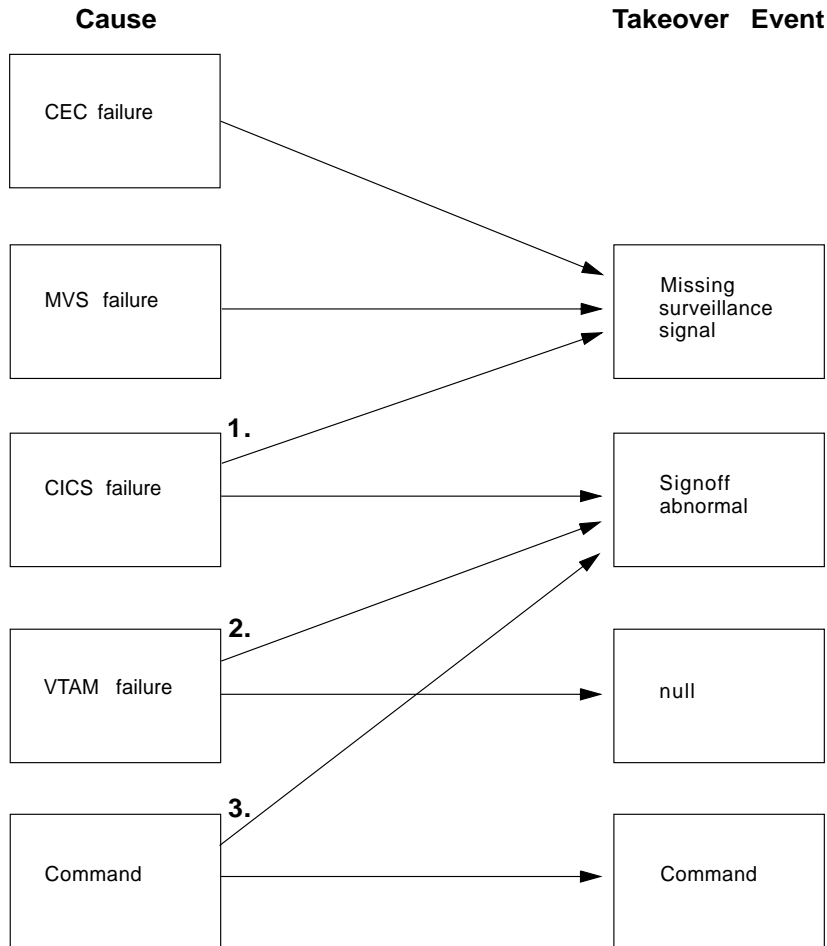


Figure 21. Causes of a takeover event

Notes:

1. For some CICS failures, CICS might be able to sign off abnormally from the CAVM. If it cannot sign off, the alternate CICS detects the missing surveillance signal.
2. A VTAM failure drives CICS user exit XXRSTAT. You code the exit to determine whether to cause a takeover by sign-off abnormal, or to allow CICS to continue normally while you restart VTAM. For information about XXRSTAT, see the *CICS/ESA 3.3 XRF Guide*.
3. The commands have already been described. They are CEBT PERFORM TAKEOVER issued to the alternate CICS, and CEMT PERFORM SHUTDOWN TAKEOVER|IMMEDIATE issued to the active CICS. Both the immediate and takeover options on the CEMT command cause the active CICS to sign off abnormally.

Table 6. Types of takeover			
Event	TAKEOVR= AUTO	TAKEOVR= MANUAL	TAKEOVR= COMMAND
Operator or program issues a CEBT command	Takeover	Takeover	Takeover
Sign-off abnormal	Takeover	Takeover	No takeover
Surveillance signal of active CICS ceases	Takeover	Conditional takeover	No takeover
Operator issues a CEMT command	Takeover	Takeover	No takeover
MVS image for active CICS fails	Takeover	Takeover	Takeover

Note: Automatic takeover after failure of the MVS image for the active CICS can occur only if the following are true:

- The active and alternate CICS are running under different MVS/ESA SP 4.1 images.
- Both MVS images are running in PR/SM LPs, on the same CEC or on the same side of the CEC if it is physically partitioned.
- Both MVS images are defined to be in the same sysplex.
- You have defined to your XCF PR/SM policy for dealing with MVS failures in a PR/SM environment (XCFPOLxx member of SYS1.PARMLIB).

If any of the above conditions are not true, MVS prompts the operator (with message IXC402D) to verify that the MVS image has failed. If the operator verifies that the MVS image has failed, CICS detects this and automatically starts XRF takeover. For further information about the effect of MVS failure, see page 99.

As shown in Table 6, the type of takeover that takes place depends on the event that occurs and the takeover option defined in the alternate CICS. The resultant action may be a takeover (which in one case is conditional), or no takeover.

What happens during a takeover

Once it has been decided that the alternate CICS should try to take over from the active CICS, a TAKEOVER request is passed to the CAVM, and the following message is written to the system console:

```
DFHXG6512I - applid : TAKEOVER REQUEST PASSED TO THE CAVM
```

In most cases this request is accepted; but it may be rejected for any of the following reasons:

- The active CICS region has already signed off normally.
- The active CICS is not the same CICS that the alternate CICS has been tracking. The CAVM detects that it is a new active, probably as a result of a restart in place. In this case, the alternate CICS is no longer able to continue its role, and a new alternate CICS should be started.
- The active and alternate CICS are on different MVS images, but **not** in a multi-MVS/ESA SP 4.1 environment, and the alternate CICS has not been

monitoring the surveillance signals of the active CICS long enough to assess the difference between the time of day clocks.

If the CAVM accepts the request, the following message is written to the console:

```
DFHXG6513I - applid : TAKEOVER REQUEST ACCEPTED BY THE CAVM
```

Once this takeover request has been accepted, an attempt by another CICS to sign on to the CAVM as an active is rejected. The alternate CICS next issues the MODIFY netname,USERVAR command to redefine the CICS application name, and begins to switch the sessions of the XRF-capable terminals.

During takeover the alternate CICS uses two different mechanisms to try to terminate the active CICS job:

1. If the active CICS is still signed on to the CAVM, the alternate CICS uses the surveillance mechanism to try to pass a “takeover-requested” message to the active, including a dump-or-no-dump indicator. If the active receives the message, it responds by issuing abend U206, and eventually signs off abnormally from the CAVM.
2. If the active job is still executing, the alternate CICS also issues a CANCEL command (prefixed by a JES routing command in a multi-MVS environment). The CANCEL is issued in case the active CICS is unable to respond to the request by the alternate CICS to take over.

Note: In some cases the CANCEL may take effect after the active CICS has already issued the abend U206, in which case CICS issues the message:

```
DFHSR0612 applid Abend recovery has been entered by same task.
```

The alternate CICS next starts to process the command list table. The alternate CICS asks JES periodically about the status of the active CICS.

Checking for termination of the active CICS

The alternate CICS asks periodically about the status of the active CICS. If the active CICS has terminated, the next phase (see “Completing the takeover” on page 103) can start immediately. If the active CICS is still executing, the alternate CICS continues asking about the status until either:

- The JESDI time interval expires, in which case the operator is asked to intervene
- CICS determines that the active CICS has terminated.

Action on takeover delay—multi-MVS/ESA SP 4.1 environment: This section describes the action on the takeover delay being exceeded, if the active and alternate CICS are running under separate MVS/ESA SP 4.1 images in the same sysplex, with the MVS images running in PR/SM LPs, on the same CEC or on the same side of the CEC if it is physically partitioned. The actions depend on the following values:

XCF parameters INTERVAL and OPNOTIFY,
XCF PR/SM policy RESETTIME and DEACTTIME
CICS JESDI and ADI.

If the active CICS fails, only the values of the CICS region initialization parameters ADI and JESDI are relevant. The ADI interval is the amount of time that the alternate CICS waits after disappearance of the surveillance signal of the active

CICS before the alternate CICS takes action. Subsequently, if a takeover is instigated, the JESDI interval is the amount of time that the alternate CICS waits for the active CICS to terminate before it prompts the operator to intervene. If JES responds that the active CICS has terminated in this intervening period, takeover continues. However, if JES continues to respond that the active CICS is still executing, the alternate CICS issues the following message to the console:

```
DFHXG6582I applid UNABLE TO DETERMINE STATUS OF JOB jesno,jobname
                ON MVS SYSTEM mvsname (sid)
```

The operator is then expected to determine why JES still believes that the active CICS is still running. The operator can either cancel the active CICS or restart JES to fix the problem. Either way, the alternate CICS eventually sees that the active CICS has terminated and issues the following message to the console:

```
DFHXG6583I - applid : TERMINATION OF JOB jesno,jobname ON MVS SYSTEM
                mvsname,smfid HAS BEEN DETECTED
```

CICS takeover then completes.

Note: You can automate XRF takeovers for active and alternate CICS on different MVS images by using an automation tool, such as NetView or TSCF, to respond when the alternate CICS issues the DFHXG6582I message, as follows:

- In response to the DFHXG6582I message, the automation tool could issue a VARY XCF,sysname,OFFLINE command on the MVS image for the alternate CICS to remove the MVS image for the active CICS from the sysplex. (This causes MVS to prompt for the operator to confirm that the MVS image is to be partitioned.)
- The automation tool could reply to the MVS prompt to confirm that the MVS image for the active CICS is to be partitioned. This causes the XCF PR/SM SYSGONE policy to be driven. The XCF PR/SM SYSGONE policy could be set up to specify that the LP of the alternate CICS is to deactivate the LP of the active CICS and acquire its resources. At this point, the alternate CICS detects that the MVS image for the active CICS is no longer in the sysplex, and automatically starts the CICS XRF takeover.

If MVS fails, the series of events mainly depends on the values specified for the XCF parameters INTERVAL and OPNOTIFY, the active PR/SM policies RESETTIME or DEACTTIME interval, and the CICS system initialization parameters ADI and JESDI.

- In a multi-MVS/ESA SP 4.1 environment, with the MVS images running in PR/SM LPs, on the same CEC or on the same side of the CEC if it is physically partitioned, the values affecting the takeover delay are shown in Figure 22 on page 100 and described in the following notes.

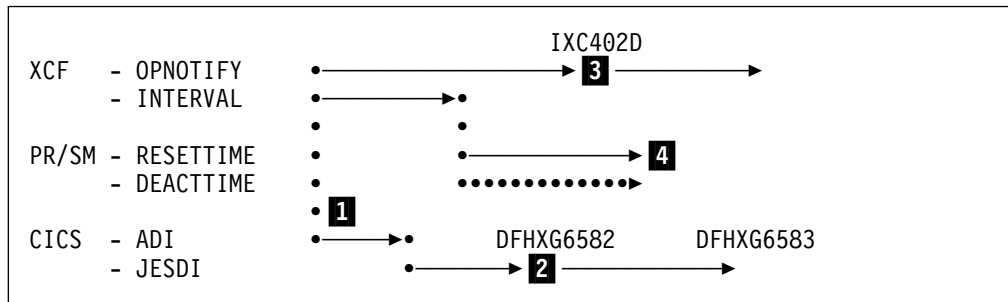


Figure 22. Values affecting takeover delay: multi MVS/ESA SP 4.1 in PR/SM LPs

Notes:

1 You can assume that the value of the XCF parameter INTERVAL starts at approximately the same time as the ADI interval in the event of an MVS failure, given that the CICS status is checked approximately every 2 seconds and MVS status approximately every 3 seconds.

After the ADI interval expires the JESDI interval starts, and similarly after the XCF INTERVAL expires the RESETTIME/DEACTTIME interval starts.

2 If the JESDI interval expires before the RESETTIME/DEACTTIME interval expires, that is before **sysplex partitioning**⁴ is started for a failed MVS image, the DFHXG6582I message is issued. However, when sysplex partitioning starts, CICS responds with a DFHXG6583I message to the console and takeover proceeds.

To avoid the DFHXG6582I and DFHXG6583I messages, you are recommended to set the ADI plus JESDI (total) interval to greater than the XCF INTERVAL plus the value specified for your XCF PR/SM policy RESETTIME or DEACTTIME interval.

3 If the OPNOTIFY interval is less than the INTERVAL plus RESETTIME/DEACTTIME interval then XCF issues message IXC402D:

```
IXC402D PRIMARY LAST OPERATIVE AT hh.mm.s REPLY DOWN IF MVS IS DOWN
OR INTERVAL=SSSSS TO SET A REPROMPT TIME.
```

but this is canceled automatically when sysplex partitioning is performed.

To avoid XCF message IXC402D in this environment, we recommend that you set the XCF OPNOTIFY value to greater than the sum of XCF INTERVAL plus that specified for XCF PR/SM policy RESETTIME or DEACTTIME.

4 When the XCF PR/SM policy RESETTIME/DEACTTIME interval expires, sysplex partitioning begins.

- In a multi-MVS/ESA SP 4.1 (or later) environment, with the MVS images **not** running in PR/SM LPs on the same CPC or the same side of the CPC if it is physically partitioned, the values affecting the takeover delay are shown in Figure 23 on page 101 and described in the following notes.

⁴ **Sysplex partitioning** is the process by which XCF removes an MVS image from a sysplex. This can be because an operator issues a VARY XCF,sysname, OFFLINE command, because an operator replies DOWN to the IXC402D message issued when XCF detects an MVS image failure, or automatically as a result of the XCF PR/SM policy.

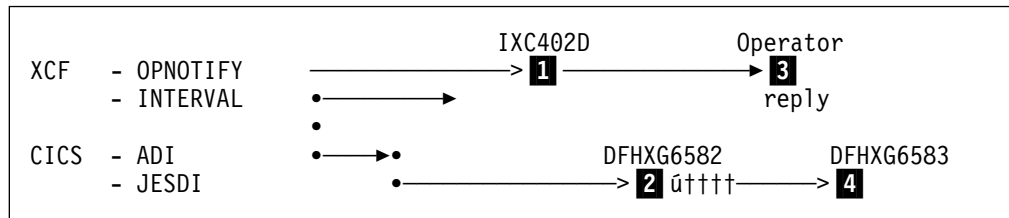


Figure 23. Values affecting takeover delay: multi MVS/ESA SP 4.1 (or later) not in PR/SM LPs

Notes:

1 After an interval expressed by the XCF OPNOTIFY interval, XCF issues message (IXC402D) to ask the operator to confirm whether a MVS image has failed.

2 If the JESDI interval expires before the operator replies that the MVS image has terminated, CICS will issue the DFHXG6582I message.

3 If an operator replies to the XCF message IXC402D that the MVS image has terminated, sysplex partitioning begins.

Note: You can use the IXC402D message as a trigger event for an automation application (such as Netview).

4 The CICS alternate detects sysplex partitioning (issues a DFHXG6583I if the DFHXG6582I has already been issued) and continues with the CICS takeover.

Action on takeover delay—single MVS/ESA SP 4.1 image: This section outlines the action on takeover, if the active and alternate CICS are running in the same MVS/ESA SP 4.1 image. In this environment, the XCF INTERVAL and OPNOTIFY values, and any active XCF PR/SM policy have no effect on any delay during CICS takeover.

If the alternate CICS is unable to determine the status of the active CICS after JESDI seconds from the end of the ADI interval, it issues the DFHXG6581 message to prompt the operator into resolving the problem that is preventing takeover from completing. The operator may resolve the problem in a number of ways, for example:

- Restart JES if it has failed. Any CANCEL commands issued while JES was not operating can then complete.
- Issue an MVS CANCEL command for the old active CICS, if it is still running, perhaps because the CLT of the alternate CICS does not contain the necessary entries to authorize it to cancel the old active CICS.
- Force the old active CICS, because a system error requiring a FORCE command has occurred.

Once the operator has completed the necessary actions, the alternate CICS detects the termination of the old active CICS, issues a DFHXG6564 message, and continues takeover.

Action on takeover delay—MVS earlier than MVS/ESA SP 4.1: This section outlines the effect of the SIT parameter JESDI interval being exceeded, if the active and alternate CICS are running under a release of MVS earlier than MVS/ESA SP 4.1.

If the active and alternate CICS are both running in the same MVS image, and the alternate CICS is unable to determine the status of the active CICS after JESDI seconds from the end of the ADI interval, it issues the DFHXG6581 message to prompt the operator into resolving the problem that is preventing takeover from completing. The operator may resolve the problem in a number of ways, for example:

- Restart JES if it has failed. Any CANCEL commands issued while JES was not operating can then complete.
- Issue an MVS CANCEL command for the old active CICS, if it is still running, perhaps because the CLT of the alternate CICS does not contain the necessary entries to authorize it to cancel the old active CICS.
- Force the old active CICS, because a system error requiring a FORCE command has occurred.

Once the operator has completed the necessary actions, the alternate CICS detects the termination of the old active CICS, issues a DFHXG6564 message, and continues takeover.

In a multi-MVS environment after the JESDI interval is exceeded, and JES replies that the active is still running, the alternate CICS issues the following message:

```
DFHXG6561D - applid : WHEN jesno jobname ENDS REPLY 'JOB' OR  
                WHEN CEC sid HAS FAILED REPLY 'CEC'
```

The alternate CICS also issues DFHXG6561D if JES is not running, or does not respond.

Note: When the active and alternate CICS are running in different MVS images, JES might continue to tell the alternate CICS that the active job is still running even though the active MVS image or its MVS system has failed. In this case, the alternate CICS cannot complete its takeover without operator intervention. Another possibility is that the active job is still running, and either never received the CANCEL command, or received it but cannot terminate because a system error needing a FORCE command has occurred.

If the MVS image for the active CICS has not failed, you must ensure that the active job really has terminated before replying 'JOB'.

If the MVS image for the active CICS appears to have failed, you must:

1. Decide whether the best way to recover is to re-IPL.
2. If possible, take a stand-alone dump.
3. If you decide to IPL, stop the processors of the failed MVS image, then perform system reset.
4. When the reset is successfully completed, reply "CEC" to the DFHXG6561 message of the alternate CICS.
5. Re-IPL the MVS image for the active CICS.

If you reply 'CEC', an internal record is kept that the MVS image, identified by its SMF system identification (SID), has failed. Other alternate CICS regions examine this record while they are performing a takeover, to try to avoid operator intervention.

Note: If you reply to DFHXG6561 prematurely, data integrity may be lost.

The alternate CICS cannot complete takeover until you reply to DFHXG6561, unless either of the following events occurs:

1. The alternate CICS receives a late reply from JES that the active job has terminated.
2. A previous reply to another DFHXG6561 message of the alternate CICS has already confirmed MVS image failure.

If either of these events occurs, a reply to DFHXG6561 is no longer required, and takeover continues. In the first case, CICS issues message DFHXG6564, and in the second case, message DFHXG6563.

Completing the takeover

When CAVM has received confirmation that the active CICS job has terminated, it notifies the alternate CICS that it may now assume the fully active role, and updates the CAVM control data set to this effect.

Emergency restart resumes, but in a multi-MVS environment, where the release of MVS is earlier than MVS/ESA SP 4.1, it is delayed until the time-of-day clock of the new active CICS has synchronized with the clock of the old active CICS.

In addition to normal emergency restart, the new active performs session cleanup for XRF-capable terminals, and re-establishes the sessions for other VTAM terminals.

Post-takeover processing

In a multi-MVS environment, additional operator activity may be required as a direct consequence of a takeover. For example:

- Physical switching of non-XRF-capable terminals is usually necessary.
- Switching of subsystems, such as DB2 (DATABASE 2).
- Restarting any CICS shared database applications.

Takeover after a command

Takeover can be initiated by either a CEBT PERFORM TAKEOVER command or a CEMT PERFORM SHUT TAKEOVER (or IMMEDIATE) command.

A CEBT PERFORM TAKEOVER command, which is used to control an alternate CICS region only, can be issued through the MVS console, or by a program. (For example, the command may be issued by another CICS region processing a CLT.) When the alternate CICS receives this command it starts to take over from the active CICS it has been tracking, without reference to the operator, *regardless of the current takeover option.*

A CEMT PERFORM SHUT TAKEOVER (or IMMEDIATE) command can be used to initiate a takeover by instructing the active CICS to shut down, and sign off abnormally from the CAVM. However, a takeover only occurs if TAKEOVR=AUTO or TAKEOVR=MANUAL is defined for the alternate CICS.

This section lists the main points of a takeover that follows from a takeover command. (These points are also shown in Figure 24 on page 104.)

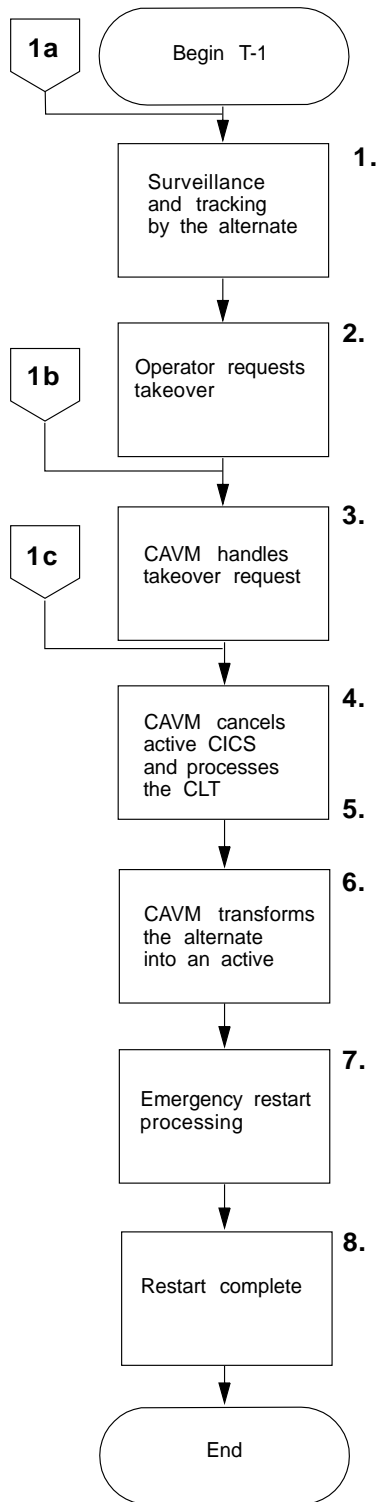


Figure 24. Chart T-1: Takeover requested by an operator-issued command

1. Surveillance and tracking running normally

While the active CICS is running normally, surveillance and tracking by the alternate CICS continues. However, although the active CICS is running normally you may need to perform a takeover for maintenance reasons. For example:

- You may need to apply service to the running CICS region; for information about applying service, see the *CICS/ESA Installation Guide*.
- If the alternate CICS is running on a different MVS image from the active CICS, you may need to perform a takeover to apply maintenance to the MVS image for the active CICS.

2. Issue the takeover command

You can initiate a takeover by either:

Issuing a CEBT PERFORM TAKEOVER to the alternate CICS, or

Issuing a CEMT PERFORM SHUT TAKEOVER to the active CICS.

When you issue a takeover command the alternate CICS attempts to start the takeover process. It does this regardless of the type of takeover defined by the system initialization parameter (TAKEOVR=AUTO|MANUAL|COMMAND), which is overridden when you issue a takeover command.

3. The CAVM handles the takeover request

It is possible that the takeover request might not be accepted by the CAVM.

For example, this could occur if:

- The active has already signed off normally from the CAVM, but the alternate CICS has not detected this yet.
- If the active and alternate CICS are running on different MVS images under a release of MVS before MVS/ESA SP 4.1, the necessary time-of-day clock difference is not available (message DFHXC6634).

If the takeover cannot proceed, the alternate CICS issues the following message:

```
DFHXG6514I applid TAKEOVER REQUEST REJECTED BY THE CAVM
```

If your takeover command is rejected, you also receive a message from the CAVM stating why. If your takeover command is accepted, you receive message DFHXG6513I accepting the request.

4. The CAVM cancels the active CICS

The alternate CICS uses two mechanisms to terminate the active CICS:

1. It uses the surveillance system to pass a “takeover-requested” message to the active.
2. It issues an MVS CANCEL command to cancel the active job.

Normally, one of these causes the active CICS to abend, with either a U206 or an S222 abend code.

If MVS rejects the CAVM’s CANCEL command, the alternate CICS issues the message:

```
DFHXG6560I applid TERMINATION COMMAND FAILED: command.
```

If, for some reason, the active CICS does not respond to the CAVM’s takeover request either, you must cancel the active CICS job manually to allow takeover to proceed.

As soon as the CAVM accepts the takeover request of the alternate CICS, the alternate CICS starts switching the sessions of the XRF-capable terminals. It also issues the MVS MODIFY netname,USERVAR command to VTAM, to change the VTAM user variable to the new specific APPLID.

For further information about the takeover process, see the *CICS/ESA 3.3 XRF Guide*.

5. The alternate CICS processes the CLT

At this point, if the alternate CICS that is taking over is a master or coordinator region in a multi-MVS environment, CEBT PERFORM TAKEOVER commands are sent to dependent regions. In addition, any write-to-operator (WTO) messages in the CLT are sent to the system console. Any other MVS commands that you include in the CLT are also processed.

6. The alternate CICS becomes the new active

When the active CICS has signed off abnormally or terminated, the CAVM stops alternate mode surveillance processing, and starts active mode surveillance processing. When CAVM detects that the active CICS job has also terminated, it notifies the alternate CICS that it may now assume the fully active role.

7. Emergency restart resumed

The emergency restart of CICS is suspended during transient data initialization pending the takeover. When the alternate CICS becomes the active CICS, emergency restart is resumed.

In addition to normal emergency restart, the new active performs session cleanup for XRF-capable terminals, and re-establishes the sessions for other VTAM terminals.

Under a release of MVS earlier than MVS/ESA SP 4.1, if the new active is running in a different MVS image from the old active, time-of-day clock synchronization takes place, and the following messages are issued:

```
DFHXC6682I applid XRF CLOCK SYNCHRONIZATION STARTED
```

Emergency restart processing is delayed while clock synchronization is in progress. Once it has completed the following message is issued:

```
DFHXC6683I applid XRF CLOCK SYNCHRONIZATION ENDED
```

In a multi-MVS/ESA SP 4.1 environment, all MVS images in the same XCF sysplex are guaranteed to be synchronized, therefore CICS does not have to synchronize the time-of-day clock during emergency restart, thus reducing the delay on takeover.

8. Emergency restart complete

When the restart is complete, control is given to CICS and transaction processing can resume.

Takeover after abnormal sign-off

If the active signs off abnormally from the CAVM, for whatever reason, and the takeover option is **not** TAKEOVR=COMMAND, the alternate CICS initiates a takeover.

This section lists the main points of a takeover that follows when the active signs off abnormally from the CAVM. (These points are also shown in Figure 25 on page 107.)

1. Surveillance and tracking running normally

While the active is running normally, surveillance and tracking by the alternate CICS continues, and the alternate CICS remains in standby mode.

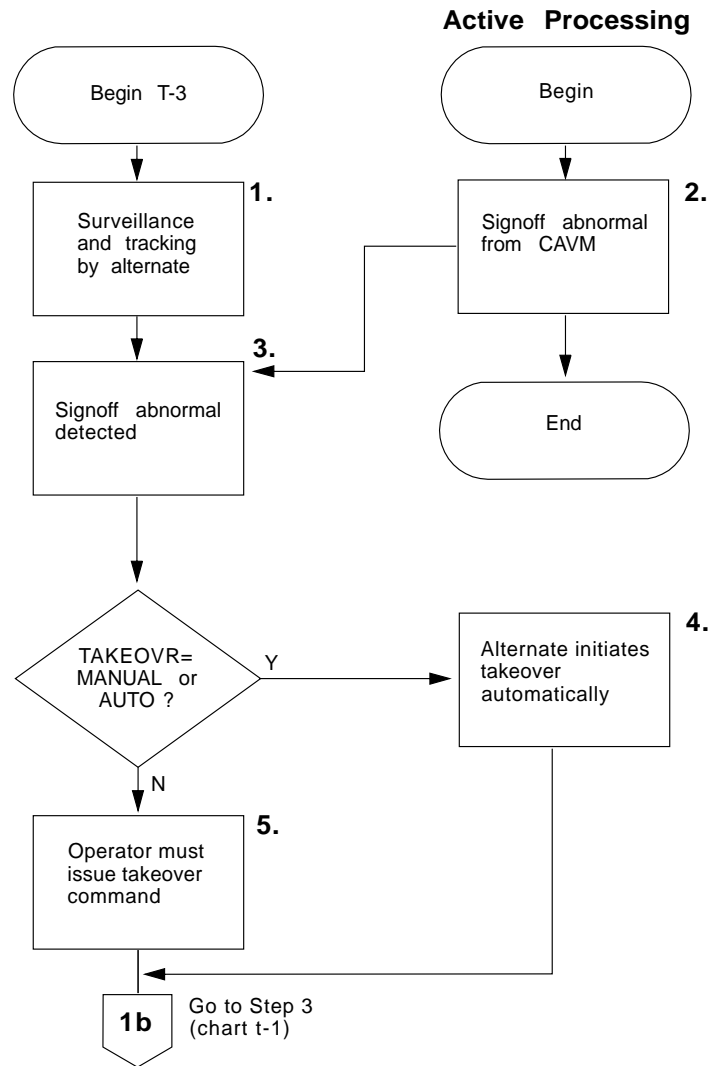


Figure 25. Chart T-3: Takeover following abnormal sign-off

2. The active CICS signs off abnormally

The active CICS signs off abnormally with the following messages:

```
DFHXG6408I applid SIGNING OFF ABNORMALLY FROM THE CAVM
DFHXG6409I applid SIGN OFF ABNORMAL FROM THE CAVM ACCEPTED
```

3. Abnormal sign-off of the active CICS is detected

The alternate CICS detects the abnormal sign-off and issues the message:

```
DFHXG6511 applid SIGN OFF ABNORMAL FROM THE CAVM DETECTED
```

The action that the alternate CICS takes next depends on the takeover option that you specified on the TAKEOVR system initialization parameter.

4. Takeover option is AUTO or MANUAL

If you specified TAKEOVR=AUTO, or TAKEOVR=MANUAL, the alternate CICS initiates a takeover, and issues the following message:

```
DFHXG6512I applid TAKEOVER REQUEST PASSED TO THE CAVM
```

The process then continues as described for the takeover after a command; see step 3 on page 105.

5. Takeover option is COMMAND

If you specified TAKEOVR=COMMAND, you must determine the cause of the problem on the active, and if it cannot be resolved, issue the takeover command. The process then continues as described for the takeover after a command; see step 3 on page 105.

Note: If a new active CICS signs on to the CAVM before the alternate CICS can complete the takeover from a failed active, the alternate CICS signs off normally from the CAVM, and terminates normally. The alternate CICS terminates because it cannot be sure that its own resource definitions correspond with those of the new active CICS. This situation is indicated by the following CAVM message:

```
DFHXC6631I applid TAKEOVER REJECTED BECAUSE LAST ACTIVE INSTANCE  
NUMBER DOES NOT MATCH THAT SPECIFIED
```

When this happens, start a new alternate CICS to continue with XRF operation.

Takeover after missing surveillance signal

Takeover can occur if the alternate CICS detects that active CICS surveillance signals have ceased

If the alternate CICS detects that the surveillance signals of the active CICS have ceased, it displays the following message, as soon as the alternate delay interval (ADI) has elapsed:

```
DFHXG6516I - applid : APPARENT FAILURE OF ACTIVE CICS DETECTED.
```

After issuing this message, the action taken by the alternate CICS depends on its current takeover option. If the current takeover options is TAKEOVR=COMMAND, the alternate CICS does **not** initiate a takeover. If the current takeover option is TAKEOVR=AUTO or TAKEOVR=MANUAL, takeover continues as described in this section.

This section lists the main points of a takeover that follows from a missing surveillance signal. (These points are also shown in Figure 26 on page 109.)

1. Surveillance and tracking running normally

While the active is running normally, surveillance and tracking by the alternate CICS continues. As long as the alternate CICS can detect the surveillance signal of the active CICS, the alternate CICS remains in standby mode.

2. The surveillance signal of the active CICS disappears

If the alternate CICS fails to detect the surveillance signal of the active CICS, the following message is issued at the console after the ADI interval has expired:

```
DFHXG6516I applid APPARENT FAILURE OF ACTIVE CICS DETECTED
```

The action that the alternate takes next depends on the release of MVS that CICS is running under. If the alternate CICS is running under a release of MVS earlier than MVS/ESA SP 4.1, or the active and alternate MVS images are not in the same sysplex, the sequence of events depends on the takeover option that you specified on the CICS system initialization parameter TAKEOVR. Otherwise, regardless of the takeover option that you specified on TAKEOVR, the processing continues as described in "Takeover after MVS failure (MVS/ESA SP 4.1)" on page 111.

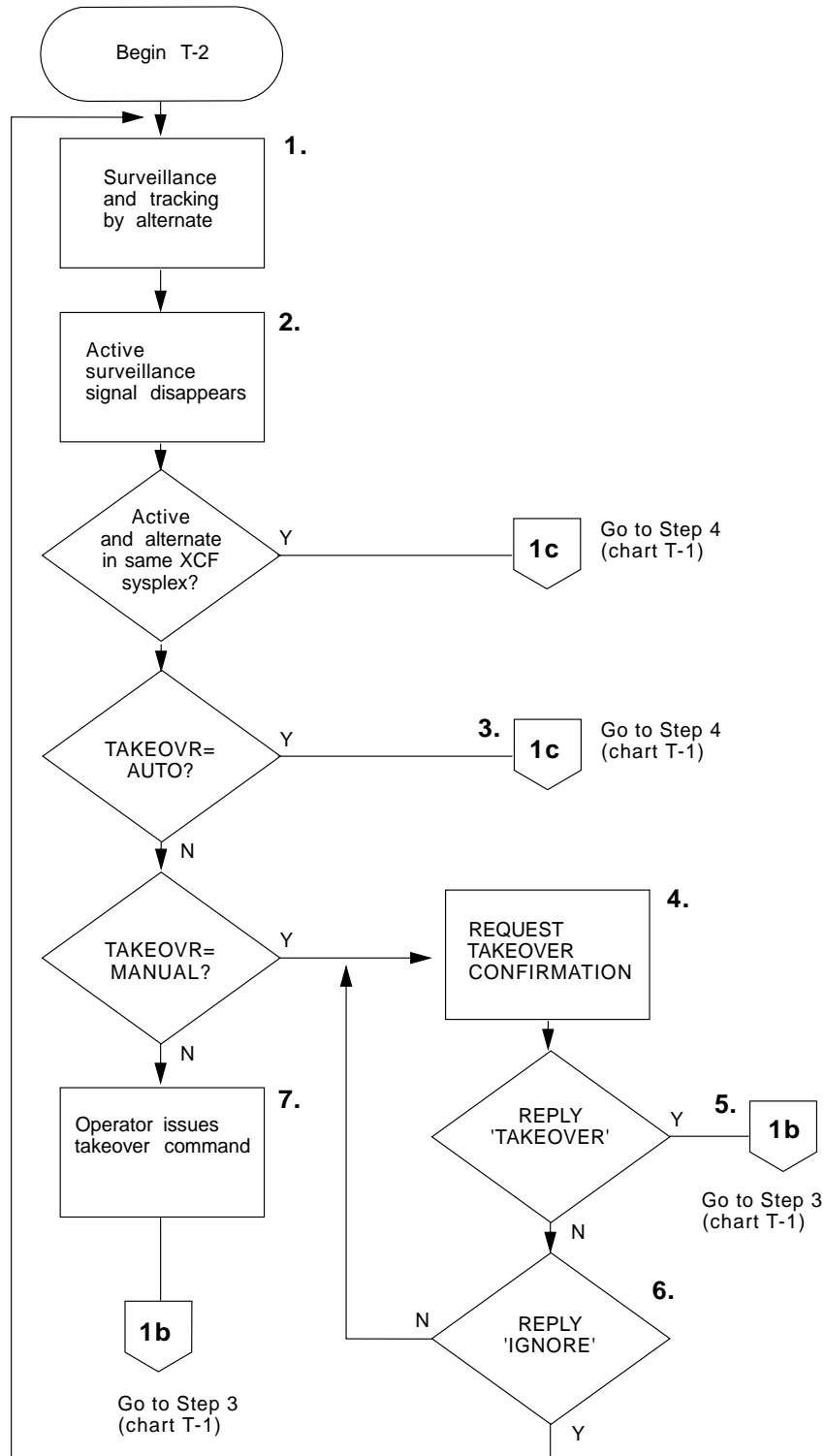


Figure 26. Chart T-2: Takeover after missing surveillance signal

3. Takeover option is AUTO

If you specified TAKEOVR=AUTO, then the CAVM automatically initiates a takeover. The process then continues as described in the takeover after a command; see step 4 on page 105.

4. Takeover option is MANUAL

If you specified TAKEOVR=MANUAL, then the CAVM issues a WTOR message requesting the operator to state what action to take:

```
nn DFHXG6518A applid APPARENT FAILURE OF ACTIVE CICS DETECTED.  
REPLY TAKEOVER OR IGNORE
```

After issuing DFHXG6518A, the action taken by the alternate CICS depends on your response to the messages displayed on the console. If you can repair the active CICS, perhaps by replying to an outstanding message, then reply IGNORE to the message. If the active CICS resumes normally, and the alternate CICS detects the surveillance signal of the active CICS, it issues the following message:

```
DFHXG6517I applid Recovery of active CICS detected.
```

If you cannot repair the active CICS, then reply TAKEOVER, in which case the alternate CICS issues a takeover request, and displays message DFHXG6512I. What happens following the takeover request is discussed under “What happens during a takeover” on page 97.

However, while the alternate CICS is waiting for your reply, it may detect another (overriding) event. For example:

- If the surveillance signal resumes before you reply to DFHXG6518, then the following messages are sent to the system console:

```
DFHXG6517I applid Recovery of active CICS detected.  
DFHXG6519I applid The reply to message DFHXG6518 is assumed to be IGNORE.
```

- If the active CICS signs off normally before you reply, then the alternate CICS terminates normally after sending the following messages:

```
DFHXG6507I - applid SIGN OFF NORMAL OF ACTIVE CICS FROM THE CAVM DETECTED  
DFHXG6519I applid The reply to message DFHXG6518 is assumed to be IGNORE.
```

- If the active CICS signs off abnormally before you reply, then the alternate CICS initiates a takeover after issuing the following messages:

```
DFHXG6511I - applid SIGN OFF ABNORMAL OF ACTIVE CICS FROM THE CAVM DETECTED  
DFHXG6519I applid The reply to message DFHXG6518 is assumed to be IGNORE.
```

5. Operator reply is ‘takeover’

If you reply ‘takeover’, the alternate CICS proceeds to carry out a takeover in the same way as if you had issued a CEBT PERFORM TAKEOVER command. The process then continues as described in the takeover after a command; see step 3 on page 105.

6. Operator reply is ‘ignore’

If you reply ‘ignore’, the alternate CICS assumes either that the active CICS recovers from the apparent failure or that it is restarted. It continues running as an alternate CICS without an active partner, monitoring through the CAVM to see if an active is started, or the old active resumes.

7. Takeover option is COMMAND

If you specified TAKEOVR=COMMAND, you must determine the cause of the problem on the active, and if it cannot be resolved, issue the takeover command. The process then continues as described in the takeover after a command; see step 3 on page 105.

Takeover after MVS failure (MVS/ESA SP 4.1)

With MVS/ESA SP 4.1, if there is an MVS failure, XRF takeover to the alternate CICS can occur regardless of the current takeover option.

Takeover completes without any operator intervention if the active and alternate CICS are running under different MVS/ESA SP 4.1 images in the same sysplex, and the MVS images are running in PR/SM LPs, on the same CEC or on the same side of the CEC if it is physically partitioned.

The operator still has to intervene if the CICS regions are running under different MVS/ESA SP 4.1 images, in the same sysplex, but the MVS images are not running in PR/SM LPs, or are running in PR/SM LPs on different CECs or on different sides of the same CEC if it is physically partitioned. (XCF itself asks the operator to verify that the system has failed, and no CICS WTOR is issued.)

If the CICS are not running under MVS/ESA SP 4.1, takeover does not occur without operator intervention on MVS failure.

This section lists the main points of a takeover that follows from the MVS image for the active CICS failing.

Note: This type of recovery occurs only if the active and alternate CICS are running under different MVS/ESA SP 4.1 images in the same sysplex.

1. Surveillance and tracking running normally

While the active is running normally, surveillance and tracking by the alternate CICS continues. As long as the alternate CICS can detect the surveillance signal of the active CICS, the alternate CICS remains in standby mode.

2. The MVS image for the active CICS fails

If the alternate CICS fails to detect the failure of the MVS image for the active CICS before the XCF PR/SM policy RESTTIME or DEACTTIME interval expires, that is, before sysplex partitioning starts, it issues the DFHXG6582I message to the console to prompt the operator. CICS continues to monitor the state of the old CICS active until it detects termination of either the active CICS or its MVS. For either event, it issues the DFHXG6583 message to the console.

Both the DFHXG6582I and DFHXG6583I messages can be avoided if the JESDI value is set to a value greater than XCF INTERVAL plus the XCF PR/SM policy interval RESETTIME or DEACTTIME.

3. Emergency restart resumes

Emergency restart completes.

What happens if the alternate CICS fails

The active CICS may detect that:

- The surveillance signal for the alternate CICS is overdue, the time-limit being defined by the SIT option PDI.
- The alternate CICS has signed off abnormally (because it has failed).

If the surveillance signal of the alternate CICS stops, the active CICS issues the message:

```
DFHXG6416I - applid APPARENT FAILURE OF ALTERNATE CICS DETECTED
```

If the alternate CICS signs off abnormally, the active CICS issues the message:

```
DFHXG6411I applid Sign off abnormal from the CAVM detected.
```

You can use the overseer to restart a failed alternate CICS. However, if you need to determine the nature of the failure, you might need to delay its restart.

Operating an XRF overseer

This section describes how to use the CICS-supplied program DFH\$AXRO, the sample overseer. You can use the overseer in two different types of situation:

1. While CICS is running normally, to monitor the active or alternate status of CICS regions running with XRF=YES
2. During failures, to restart failed regions in place (probably failed dependent regions in a multi-MVS environment).

For programming information about the CICS overseer, and how to add your own code to the sample program provided (so that the overseer performs the functions that you want it to) see the *CICS/ESA Customization Guide*.

The overseer sends messages to the system console.

MVS MODIFY commands to control the overseer

You can use the following MVS MODIFY commands to communicate with the overseer. You can abbreviate valid commands to only the initial letter.

Display

Displays the current status of all CICS regions known to the overseer.

Restart

Enables/disables restart-in-place for all CICS regions known to the overseer.

Snap

Takes a snap dump of the overseer.

End

Terminates the overseer.

The first two commands cover the major two functions of the overseer, namely display and restart-in-place.

Display command

You can monitor the status of a CICS region running with XRF by using the display command to provide a multi-line WTO (MLWTO) to show the last known state of each of the CICS regions defined to the overseer. The display shows a title line and underneath this a line for each active and alternate pair of CICS regions. Each display line includes:

- The generic applid
- The jobname of the active CICS

- The specific applid of the active CICS
- The SMF ID of the CPU on which the active is executing, or was last known to be executing
- The current, or last known, status of the active CICS. The status can be:
 - ACT** The active CICS is signed on to the CAVM and running, or about to run, the active CICS workload.
 - SOFN** The active has signed off normally from the CAVM, because of a normal shutdown.
 - SOFA** The active has signed off abnormally from the CAVM.
 - INCA** The old active CICS has signed off abnormally from the CAVM, but has not yet terminated. The alternate CICS is taking over from the active CICS, and the alternate CICS is now partially active.
 - TKIP** A takeover is in progress. The alternate CICS of this active CICS region is attempting a takeover.

An 'X' following any of these states indicates that the job is executing. No 'X' indicates that the job is not executing.

- The jobname of the alternate CICS
- The specific applid of the alternate CICS
- The SMF ID of the CPU on which the alternate CICS is executing (or was last known to be executing)
- The current, or last known, status of the alternate CICS. The status can be:
 - BACK** Alternate CICS signed on to the CAVM and running normally.
 - OLD** The information displayed refers to out-of-date information about the CICS region that was the alternate CICS until a recent takeover. The CICS region is now the current active and the information displayed for the active CICS is marked as OLD until a new alternate CICS is signed on and running normally.
 - SOFA** The alternate CICS has signed off abnormally from the CAVM.
 - SOFN** The alternate CICS has signed off normally from the CAVM, either because of a normal shutdown by the active CICS or because of a CEBT PERFORM SHUT command issued to the alternate CICS.
 - TKOV** A takeover has been initiated by this alternate CICS.
 - UNKN** The overseer has no current information about the status, which was in an intermediate state when the display command was processed. Reissue the command to replace UNKN by a more meaningful status value.

An 'X' following any of these states indicates that the job is executing. No 'X' indicates the job is not executing.

Restart-in-place command

The overseer sample program is designed to restart failed active and alternate CICS regions automatically under certain circumstances. It achieves this either by submitting a job or by issuing an MVS START command.

You use the 'R' command to enable or disable the ability of the overseer to attempt to restart a failed region. After initialization, restart is enabled. If you issue the 'R' command, you disable restart processing. If you issue it again, you re-enable restart processing, and so on. Each time you issue the 'R' command, the overseer displays a message to inform you of the effect. Restart processing is enabled/disabled for all jobs known to the overseer.

In brief, restart-in-place is used when you would prefer to restart a region in the same MVS image, instead of allowing an alternate CICS in a different MVS image to take over. A dependent region in a multi-MVS environment is a good example, but you might also prefer to restart in place, if possible, regions that you would normally expect to be taken over, like a VTAM terminal-owning region. For each region, you have to weigh up the possibility that, if successful, a restart in place could involve less interruption to service than a takeover to another MVS image. For programming information about how to define each region to the overseer, and of the circumstances in which it attempts to restart in place, see the *CICS/ESA Customization Guide*.

How the overseer restarts failed regions in place: If the failed region was started originally as a started task (STC), the DFH\$AXRO overseer restarts it as a started task; if the failed region was started as a job, the overseer restarts it as a job.

- To restart a failed region that was executing as a started task, DFH\$AXRO issues the following command:

```
START xxxxxxxx,CSTART=AUTO|STANDBY
```

The variable xxxxxxxx is the name of a procedure provided by you. It is the jobname of the started task region that has failed, and becomes the jobname of the region restarted by the overseer. The CSTART value tells CICS whether it should initialize as an active CICS (AUTO), or as an alternate CICS (STANDBY).

You need two MVS procedures, suitably coded, for each region (an active and alternate CICS pair has two jobs, either of which could be the active or the alternate CICS), with the CICS override START=&CSTART as a PARM parameter in both. For example:

```
EXEC PGM=DFHSIP,REGION=2048K,PARM='... ,START=&CSTART,...'
```

- To restart a failed region that was executing as a job, DFH\$AXRO issues the following command:

```
START DFHCRST,CJOB=xxxxxxx,CSTART=AUTO|STANDBY
```

DFHCRST is the name of a CICS procedure in CICS410.SDFHPROC, which you must tailor for your environment. The CJOB value xxxxxxxx is the jobname of the region that is being restarted. The CSTART value tells CICS whether it should initialize as an active CICS (AUTO), or as an alternate CICS (STANDBY).

DFHCRST uses a control file, DFHXROCL, to copy the failed job's JCL from a user library into a temporary partitioned data set (PDS) If you are going to use the supplied procedure, the JCL must have a 'START=xxxx' keyword in an inline SYSIN data set, starting in column one. (The procedure assumes that all CICS jobs reside in the same library, and as supplied the USRLIB parameter is coded with CICS410.SDFHINST. See the comments in the procedure for an explanation of the other parameters used.) The procedure splits the JCL and inserts the required START=AUTO or START=STANDBY keyword before adding the remaining JCL. The JCL is then submitted to the internal reader.

You must ensure correct security (RACF) authorization for the DFHCRST procedure to copy members from the JCL and CICS410.SDFHMAC libraries, and also that the submitted job has the correct RACF authority.

DFHCRST is shipped in CICS410.SDFHPROC, and DFHXROCL is shipped in CICS410.SDFHMAC.

SNAP command

This command requests that a SNAP dump be taken to the SNAPOUT dump data set.

End command

This command terminates the overseer.

CICS overseer modules

The CICS overseer comprises four modules, three of which are CICS-supplied, and a user-written module, which you generate. There is a CICS-supplied sample of a user-written overseer program, called DFH\$AXRO, in CICS410.SDFHSAMP, which you can customize and rename if you want to. The CICS overseer modules, which are all in CICS410.SDFHAUTH, are:

1. DFHWOS The XRF overseer bootstrap module
2. DFHWOSA The XRF overseer region initialization module
3. DFHWOSB The XRF overseer region service routines.

The CICS overseer modules provide a stable interface to the CAVM data sets, and to certain MVS-authorized services that the overseer program requires. The overseer program invokes those services through a CICS-supplied group of macros called the DFHWOSM macros; programming information about these is in the *CICS/ESA Customization Guide*. When the region is initialized, DFHWOSB invokes either the CICS-supplied sample overseer, or a user-written version. You supply the name as a parameter on the EXEC statement to start the overseer.

The CICS-supplied sample overseer program, DFH\$AXRO, and the three overseer modules, DFHWOS, DFHWOSA, and DFHWOSB, are compatible with releases of CICS before CICS/ESA 4.1, so can process CAVM data sets from earlier releases of CICS.

There is a sample overseer job in CICS410.SDFHINST, called DFHIVXRO, which is designed to run with the CICS IVPs when XRF=YES is coded. The rest of this chapter describes how to create an overseer data set for use with the CICS-supplied sample program, DFH\$AXRO, and discusses the job control statements you can use to run the sample overseer.

Using a DFHOSD data set to tell the overseer what to monitor

The sample overseer program is written to handle active and alternate pairs of CICS regions, and related system names. The maximum number of active and alternate CICS pairs that can be monitored by the overseer is 50.

A 'related system name' identifies those regions that cannot be considered in isolation by the overseer. The most common example of this is an MRO environment, where the overseer needs to be able to identify related regions when deciding whether to restart a failed region in place. Those regions that are identified with a common related system name must execute on the same MVS image.

The sample program determines which active and alternate CICS pairs it is monitoring from a VSAM key-sequenced data set with a ddname of DFHOSD, which contains a single entry for each active and alternate CICS pair. You create this data set and initialize it with information about active and alternate CICS pairs before you use the overseer for the first time. You also have to redefine the DFHOSD data set whenever you want to change the information that it holds.

You can use the sample job stream listed in Figure 27 on page 117 to:

1. Delete and define the DFHOSD data set
2. Initialize the DFHOSD data set with information about sample active and alternate CICS pairs.

The sample overseer program reads the DFHOSD records in key sequence and builds a table of entries, stopping when the first 20 entries have been read.

Each active and alternate CICS pair is known by its generic applid on this data set. The records in the DFHOSD data set contain the information described in Table 7.

Field and start position	Description
Record key, starting in column 1	A 12-byte key, containing the 4-byte value "GNbb" followed by the 8-byte generic applid of the active and alternate CICS pair. (The applid starts in column 5.)
Related system identifier, starting in column 14.	An optional 8-byte name, to identify related regions (for example to identify related MRO regions).
Control data set ddname, starting in column 23.	The 8-byte ddname, as on the DD statement in the JCL to run the overseer, that identifies the CAVM control data set associated with this generic applid.
Message data set ddname, starting in column 32.	The 8-byte ddname, as on the DD statement in the JCL to run the overseer, that identifies the CAVM message data set associated with this generic applid.
Restart indicator, starting in column 41.	A 1-byte restart-in-place indicator, to show whether a region can be restarted in place. The only value that prevents an attempt to restart in place is the letter N.

The data structure (DSECT) of the DFHOSD data set entries is provided in member DFH\$XRDS of CICS410.SDFHSAMP. You can use the job in Figure 27 to create and initialize DFHOSD.

```
//DEFOSD JOB accounting information,CLASS=A,
//          MSGLEVEL=(1,1),MSGCLASS=A,NOTIFY=userid
//DEFDS EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
DELETE ('CICS410.DFHOSD') PURGE CLUSTER
DEFINE CLUSTER (TRACKS(1,1) -
NAME (CICS410.DFHOSD) -
SHAREOPTIONS(2) -
VOLUMES (vol1id)) -
DATA -
(RECORDSIZE(80,80) -
CONTROLINTERVALSIZE(2048) -
NAME (CICS410.DATA.DFHOSD) -
KEYS(12,0)) -
INDEX -
(NAME(CICS410.INDEX.DFHOSD)) -
/*
//INITOSD EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
/* COLUMN
/* 1 5 14 23 32 41
/* GN genappl relsysid ddnxrctl ddnxrmsg ind
/*
//SYS01 DD *
GN DBDCCICS CTLDBDC MSGDBDC
GN DBDCMR01 MROSYS CTLMR01 MSGMR01 N
GN DBDCMR02 MROSYS CTLMR02 MSGMR02
GN DBDCMR03 MROSYS CTLMR03 MSGMR03
GN DBDCMR04 MROSYS CTLMR04 MSGMR04
GN DBDCMR05 MROSYS CTLMR05 MSGMR05
/*
//SYSIN DD *
REPRO INFILE ( SYS01 ) -
OUTDATASET (CICS410.DFHOSD)
/*
```

Figure 27. Sample job to create and initialize the DFHOSD data set

Notes:

1 This entry is for an XRF system that is not associated with any other CICS region, and does not require a related system identifier. This entry is suitable to use with the XRF IVPs supplied with CICS; for information about these IVPs, see the *CICS/ESA Installation Guide*.

2 The last five entries are for three related MRO regions which are identified by the related system name of MROSYS. One of these is not allowed to restart in place, indicated by the letter N. The last two entries are for related MRO regions running CICS/ESA 3.1.1 and CICS/MVS 2.1. CAVM data sets for regions running at all releases from CICS/MVS Version 2 can be processed by the CICS/ESA 4.1 overseer.

Job control statements to run the CICS overseer

The job shown in Figure 28 is an example of the job control statements you need to run the CICS overseer in an XRF environment. The accompanying notes explain some of the things you must observe when writing a job stream to run a CICS overseer.

There is a CICS job supplied on the distribution tape with the pregenerated system, intended for use with the CICS IVPs. It is copied to CICS410.SDFHINST as part of the installation process; for information about the installation process, see the *CICS/ESA Installation Guide*.

```
//OVERSEER JOB  accounting info,name,
//              MSGCLASS=A,MSGLEVEL=(1,1),
//              CLASS=A,NOTIFY=userid
//*****
/* This job is an example of the job control statements needed   *
/* to run the CICS overseer in an XRF environment.               *
//*****
//RUNOVSR  EXEC  PGM=DFHWOS,REGION=500K,TIME=1440,
//              PARM='CEC=nn,CYTIM=nnnn,IOPN=DFH$AXRO'           1
//STEPLIB  DD  DSN=CICS410.userlib,DISP=SHR                       2
//DFHLIB   DD  DSN=CICS410.SDFHAUTH,DISP=SHR                     3
//DFHOSD   DD  DSN=CICS410.DFHOSD,DISP=SHR                       4
/* CICS/ESA 4.1 CAVM control and message data sets
//CTLDBDC  DD  DSN=CICS410.DFHXRCTL,DISP=SHR                     5
//MSGDBDC  DD  DSN=CICS410.DFHXRMSG,DISP=SHR
//CTLMR01  DD  DSN=CICS410.MR01.DFHXRCTL,DISP=SHR
//MSGMR01  DD  DSN=CICS410.MR01.DFHXRMSG,DISP=SHR
//CTLMR02  DD  DSN=CICS410.MR02.DFHXRCTL,DISP=SHR
//MSGMR02  DD  DSN=CICS410.MR02.DFHXRMSG,DISP=SHR
//CTLMR03  DD  DSN=CICS410.MR03.DFHXRCTL,DISP=SHR
//MSGMR03  DD  DSN=CICS410.MR03.DFHXRMSG,DISP=SHR
/* Earlier release CICS CAVM control and message data sets
//CTLMR04  DD  DSN=CICS311.MR04.DFHXRCTL,DISP=SHR
//MSGMR04  DD  DSN=CICS311.MR04.DFHXRMSG,DISP=SHR
//CTLMR05  DD  DSN=CICS210.MR05.DFHXRCTL,DISP=SHR
//MSGMR05  DD  DSN=CICS210.MR05.DFHXRMSG,DISP=SHR
//SNAPOUT  DD  SYSOUT=A
//SYSUDUMP DD  SYSOUT=A
//SYSABEND DD  SYSOUT=A
//SYSPRINT DD  SYSOUT=A
//
```

Figure 28. Sample job to run the CICS overseer

Notes:

1 The full syntax of the PARM option on the EXEC statement for module DFHWOS is as follows:

```
// EXEC PGM=DFHWOS,
//       PARM='IOPN=iopgmname[, {CEC={3|nn}}[, CYTIM={60|nnnn}]'
```

The parameters are:

IOPN=iopgmname

IOPN is the installation's overseer program name. For the CICS-supplied sample program it is DFH\$AXRO, as shown in the sample job stream. If you write your own overseer program, code its name here.

CEC=

Code this with the maximum number of concurrent program calls the overseer can make, in the range 3 through 24. If you omit this parameter, the default is 3.

CYTIM=[60|nnnnn]

This parameter specifies the time interval, or 'cycle time' (CYTIM), that is allowed for a CAVM data set read to complete. Each read request is tried three times before failing. The time that you specify is allowed for each attempt, and it must be in the range of 20 to 32767 seconds. The default value is 60 seconds.

2 The CICS330.userlib library contains the user-written or CICS-supplied sample overseer program, and does not need to be APF-authorized. However, as an overseer program is able to issue arbitrary MVS commands you should ensure that the library is RACF-protected for security reasons. Further, the DFHWOS module must run as an APF-authorized program, and must be installed in an APF-authorized library. If DFHWOS is not installed in the LPA or a library in the MVS linklist, you must ensure that all libraries in the STEPLIB concatenation are APF-authorized. If you want to place the overseer program, which is not APF-authorized, into the STEPLIB concatenation, you must install it into an APF-authorized library, which may violate your security and integrity rules. Therefore, you should install the DFHWOS module in an APF-authorized library in the MVS linklist or in the LPA.

3 The CICS-supplied overseer modules, DFHWOSA and DFHWOSB, are generated into CICS410.SDFHAUTH, which must be an authorized library. These modules are loaded from DFHLIB when you run the overseer.

4 The DFHOSD data set provides the control information for the CICS sample overseer program, DFH\$AXRO. If you choose the same method of providing control information for your own overseer, you must ensure that the data set is initialized before use. There is a sample job in Figure 27 on page 117 which shows you how to create and initialize the DFHOSD data set for use by DFH\$AXRO.

5 The ddnames used on the DD statements for the CAVM control and message data sets must be those defined in the DFHOSD data set for each generic applid. See the descriptions of the DFHOSD record fields shown in Table 7 on page 116.

The control and message data sets identified by the DD statements in this job must all have been created before you attempt to run the overseer job, but not necessarily initialized by their respective CICS regions. However, if they have not yet been initialized when you start the overseer, you cannot receive any information about the CICS regions they refer to.

The last two pairs of CAVM data sets are for CICS regions running at CICS/ESA 3.1.1 and CICS/MVS 2.1. These regions are also defined in the CICS/ESA 4.1 DFHOSD data set (see Figure 27 on page 117).

A run-time example

During its execution, the overseer program keeps the operator informed of its operations and of any errors by issuing messages to the MVS console (see Figure 29). The first line of every message from the overseer begins with the characters +OVERSEER:. A complete list of these messages, with explanations, is provided in the prolog of the source code of the sample overseer program.

```
18.00.49 JOB 6840 IEF403I OVERSEER - STARTED - TIME=18.00.49
18.00.49 . +OVERSEER: ENTERING INITIALIZATION
18.00.50 . +OVERSEER: DFHOSD DATASET OPENED
18.01.12 . +OVERSEER: DFHOSD DATASET CLOSED
18.01.12 . +OVERSEER: INITIALIZATION COMPLETE
:
:
modify overseer,R
:
:
18.01.29 . +OVERSEER: RESTART IN PLACE IS NOW DISABLED
:
:
modify overseer,R
:
:
18.01.36 . +OVERSEER: RESTART IN PLACE IS NOW ENABLED
:
:
modify overseer,D
:
:
18.01.41 . +OVERSEER: CPU CEC5 DISPLAY
          +GEN-APP ACT-JOB ACT-APP ACPU A-ST BKP-JOB BKP-APP BCPU B-ST
          DBDCCICS DBCICSJ1 DBDCCIC1 CEC5 ACT X DBCICSJ2 DBDCCIC2 CEC4 SOFN
:
:
modify overseer,S
:
:
18.02.19 . $HASP375 OVERSEER ESTIMATED LINES EXCEEDED
18.02.23 . +OVERSEER: SNAP DUMP TAKEN
Job DBCICSJ1 fails and no alternate CICS is available:
18.03.04 . +OVERSEER: JOB IS BEING RESTARTED
18.03.04 . START DFHCRST,CJOB=DBCICSJ1,CSTART=AUTO (BY IOP) 1
18.03.04 . START DFHCRST,CJOB=DBCICSJ1,CSTART=AUTO
:
:
modify overseer,E
:
:
18.03.38 . +OVERSEER: TERMINATION COMPLETE
18.03.39 . IEF404I OVERSEER - ENDED - TIME=18.03.39
```

Figure 29. The overseer run-time example

1 The overseer has two ways of restarting failed regions in place; these are described on page 114. If the job were being resubmitted as a started task, rather than as a job, the restart messages would be as follows:

```
18.03.04 STC 6840 START DBCICSJ1,CSTART=AUTO (BY IOP)
18.03.04 STC 6840 START DBCICSJ1,CSTART=AUTO
```

The message (BY IOP) tells you that restart has been initiated by the installation overseer program (IOP).

Chapter 7. Operating CICS from a console device

You can operate CICS from a **console device**⁵. In particular, you can use the console device for CICS master terminal functions, to control CICS terminals or to control several CICS regions in conjunction with multiregion operation. Normal operating-system use of the console device is not inhibited, and CICS supports multiple console devices where present.

You can invoke CICS transactions from a console device by using the MVS MODIFY command (F for short), and other CICS operators can communicate with the console device operator.

Note: The CEDA transaction can be used from a console device only to INSTALL resource definitions. The sample programs cannot be executed from a console device.

To communicate with an alternate CICS in an XRF environment, you must use the CEBT command, and issue it from a console device.

If you are running CICS under MVS/ESA SP 4.1 or later, you can use TSO CLIST processing to issue sequences of CICS commands. You can also use an automated process, such as NetView, to issue CICS commands as from a console device. To associate command responses with the originating command from an automated process, you must add a command and response token (CART) to the originating command. CICS returns this CART in all write-to-operator (WTO and WTOR) macros issued in response to the command.

If you issue the MVS command 'd consoles', this displays a list of console devices and their numbers (before MVS/ESA SP 4.1) or their names (for MVS/ESA SP 4.1 or later).

You can define a console device to be used for putting MODIFY commands into your job stream. For more information about putting commands into job streams, see "Using JCL to initiate CICS commands" on page 122.

For information about defining console devices to CICS, see the *CICS/ESA System Definition Guide*. For further information about defining console devices to MVS, see the *MVS/ESA Initialization and Tuning Guide* manual.

⁵ If you are running CICS under a release of MVS before MVS/ESA SP 4.1, a console device is a locally-attached system console. However, if you are running CICS under MVS/ESA SP 4.1 or later, a console device can be a locally-attached system console, a TSO user defined as a console, or an automated process such as Netview.

Entering commands from a console device

To enter a CICS command from a console device, use:

```
{MODIFY|F} cicsid,[']command[']
```

where:

cicsid is the region identifier for the CICS region. This is one of the following:

- name of the job being used to execute CICS
- name of a procedure if CICS was initiated as a started task without a qualifier
- name of the task identifier qualifier if CICS was started as a started task with a qualifier.

command is a string of data, starting with a CICS transaction identifier.

For example:

```
MODIFY DFHIVPOL,'CEMT INQUIRE TASK'
```

If a transaction started at a console device requires further input, you are prompted in the same way as a terminal operator. For more information about continuing transaction input, see "Replying to messages from transactions started at console devices" on page 125.

Entering commands from TSO

If you are running CICS under MVS/ESA SP 4.1 or later, a TSO user can enter CICS commands as above after invoking the TSO command **CONSOLE**, in either of the following formats:

```
CONSOLE SYSCMD ({MODIFY|F} cicsid,[']command['])
```

```
CONSOLE  
{MODIFY|F} cicsid,[']command[']  
END
```

When the TSO command **CONSOLE** is used, TSO checks the user for authority to issue console commands. Further, if console operator command security is active, the TSO user must be specifically authorized to issue **MODIFY cicsid**.

Using JCL to initiate CICS commands

If you have defined a console entry in your CSD as one of the following:

- **CONSOLE(00)** for CICS running under MVS before MVS/ESA SP 4.1

#

Apar PQ15423

#

Documentation for Apar PQ15423 added 23/06/98

#

- **CONSNAME(INTERNAL)** or **CONSNAME(INSTREAM)**, depending on the CICS service level and the release of MVS being used, for CICS running under MVS/ESA SP 4.1 or later,

#

#

you can submit commands to your CICS region by using JCL.

Your JCL should use the MVS MODIFY command and the job name, or task ID, of the CICS region you are addressing, followed by the CICS commands. The normal rules of JCL apply.

The following sample job shows how you might submit commands in this way.

```
//IEFBR14 JOB (accounting information),CLASS=A,MSGCLASS=A,MSGLEVEL=1,....,....  
//*  
//* Sample JOB to submit CICS commands using CONSNAME(INTERNAL)  
//*  
//IEFBR      EXEC PGM=IEFBR14  
// F CICSRUN,'CEMT INQ TER'  
// F CICSRUN,'CEMT INQ TAS'  
// F CICSRUN,'CEMT SET TER(L77C) ACQ'  
//
```

Note: If you omit the apostrophes around the CICS command, and there are sequence numbers at the end of the line, the numbers are passed to CICS as part of the command. This causes CICS to display a warning message on the console, but the command is still obeyed.

Console device messages

During both the initialization and the running of CICS, various messages appear on your console device. These are mainly for information, but in some cases may require a reply or some action from you.

The console messages may be subject to message-formatting if you have defined CICS as an MVS subsystem with console message-handling support.

Console message-formatting

You can define CICS as an MVS subsystem with support for the console message-handling facility. By using this facility, CICS can enable MVS to:

- Convert all console messages to the same format, and
- Insert into each message the applid of the sending region.

Note: The term **console message** is used for messages sent to the system console, *not* CSMT messages or the JES joblog.

The main purpose of the console message-handling facility is to ensure that all messages issued by CICS regions contain the APPLID of the CICS region issuing the message.

You specify that CICS is to use the console message-handling facility when you define CICS as an MVS subsystem (by the CICS entry in the IEFSSNaa member of the SYS1.PARMLIB library). If the message-handling facility has been defined for CICS, all messages from all CICS regions (of any release) are intercepted and reformatted to include the APPLID if either of the following is true:

- An automated-operation program, such as NetView, is active in the MVS image.
- A CICS region that supports message handling is running in the same MVS image. This includes CICS/ESA 4.1 regions, CICS/ESA 3.3, CICS/ESA 3.2.1 and CICS/MVS 2.1.2 regions with console message-handling support, and CICS/ESA 3.1.1 regions with APAR PL66570 applied.

For information about defining CICS as an MVS subsystem with support for the console message-handling facility, and about activating the facility, see the *CICS/ESA Installation Guide*.

Message format

The following examples show three messages as they appear with and without console message formatting. The examples use CICSIDC as the applid of the sending region.

- Message format without console message formatting:
DFH5730 - USER RECOVERY BEGINNING
DFH5731 - NO ACTIVE USER RECORDS ON THE SYSTEM LOG
DFH5732 - USER RECOVERY COMPLETED
- Message format with console message formatting:
DFH5730 CICSIDC USER RECOVERY BEGINNING
DFH5731 CICSIDC NO ACTIVE USER RECORDS ON THE SYSTEM LOG
DFH5732 CICSIDC USER RECOVERY COMPLETED

Advantages of message formatting

The main benefits of using console message formatting are:

- Assistance to the console operator
- Ease of automated operation by a program such as NetView.

The implementation of message formatting also:

- Allows masking of the password entered at the console during the CICS signon transaction. For example, you might enter the following command to sign on to CICS from a console:

```
F CICS,CESN USERID=HARBEN, PS=HUMMER, NEWPS=STONE
```

The passwords are then obliterated with asterisks when the command is redisplayed on the console or recorded in the system log.

```
F CICS,CESN USERID=HARBEN, PS=*****, NEWPS=*****
```
- Allows the adding of a set of MVS generic routecodes to all CICS console messages, permitting them to be sent to a defined set of consoles.
- Removes the restriction that prevents the use of the name *CICS* as the MVS jobname of a CICS region that is started with the START command.

Suppressing information-only messages

You can use the system initialization parameter MSGLVL to control the generation of messages to console devices. If you code MSGLVL=0, only critical errors or interactive messages are printed.

Replying to messages

If one or more CICS messages are followed by an associated message that requests an operator response, the earlier message or messages may have scrolled off the console screen before the response-requesting message appears. Some messages that need a reply include a preceding message number or specify a response that can be entered to display the preceding message.

If a message requests a reply but does not provide means of determining the previous messages that explain the response required, CICS retains, in the message buffer, all messages in the logically-related set, until a valid response is received to the final message. When the console displays a message that requires a response, the operator can request a display of all preceding related messages. A typical message that needs a response is:

```
DFHJC4543D date time applid Journal data set journalno
          datasetid is still not ready. Reply 'GO' or 'SUBMIT'.
```

If such a message appears, the operator can display all the preceding related messages by entering the MVS command:

```
DISPLAY R,I
```

When a valid response is received to the final message in the set, CICS deletes all the related messages from the message buffer.

Replying to messages from transactions started at console devices

If a transaction started at a console device requires further input, you are prompted in the same way as any normal terminal operator. You can continue the input in one of the following ways:

- If the transaction is conversational and uses the CONVERSE or RECEIVE command, the message from CICS will contain a reply number that must be quoted in the reply. This is described in this section.
- If the transaction is pseudo-conversational, you must enter further MODIFY commands to continue the conversation.

You respond to messages from transactions started at a console device by using the REPLY command (abbreviation R). For example:

```
REPLY 02,'datastring'
```

where 02 is the number of the message to which you are replying, and 'datastring' is your reply. If you cancel a transaction that is running at a console device, and the transaction is awaiting a reply, the outstanding reply is also canceled.

For information about using CEMT and the other CICS-provided transactions, and about entering transactions from a console, see the *CICS/ESA CICS-Supplied Transactions* manual.

If you try to communicate with an active CICS region from a console device that has not been defined to CICS, you get message DFHAC2015 saying that your console has not been defined to CICS and that your input will be ignored.

In a CICS region that has consoles and VTAM terminals, a console can remain active when CICS and VTAM are disconnected from each other. This means that you can use the console to open or close the CICS-VTAM connection without CICS being terminated.

Suppressing and rerouting messages

CICS provides a global user exit point, XMEOUT, that is invoked before a message is sent from the message domain to its destination. XMEOUT can be used to invoke an exit program to intercept messages issued by SEND MESSAGE requests, and suppress the messages, change their destination, or leave them alone.

CICS provides six sample user exit programs, DFH\$SXP1 through DFH\$SXP6, that you can use to suppress or reroute messages.

For programming information about the global user exit XMEOUT and the sample exit programs, and the user exit programming interface (XPI), see the *CICS/ESA Customization Guide*.

Sample console messages for CICS startup

Sample console messages issued when CICS starts up are given in “System console messages for CICS startup” on page 22.

Sample console messages for CICS shutdown

Sample console messages issued when CICS starts up are given in Chapter 8, “How to shut down CICS” on page 127.

Chapter 8. How to shut down CICS

This chapter

describes how to shutdown CICS. For an overview of CICS shutdown, see “Shutting down CICS” on page 13.

To shutdown CICS, you can issue the CEMT PERFORM SHUTDOWN command with appropriate options, depending on the type of shutdown that you want. You can specify any of the following shutdown options on the command, without affecting the type of shutdown performed:

Option Effect

DUMP CICS will produce a dynamic storage dump after shutdown has completed.

PLT(xx) CICS will run programs in the PLT, DFHPLTxx, during shutdown.

XLT(xx) Only those transactions listed in the XLT, DFHXLTxx, can be started after the SHUTDOWN command, and before shutdown has completed.

You can use the CEMT PERFORM SHUTDOWN command at the master terminal, or the system console.

For guidance on using the CEMT transaction, see the *CICS/ESA CICS-Supplied Transactions* manual.

Shutting down CICS normally

To shut down CICS normally, use the CEMT PERFORM SHUTDOWN command.

Note: CICS normal shutdown cannot complete until all pipes (sessions) in use for the external call interface have been closed.

What you should see

When you use the CEMT PERFORM SHUTDOWN command, CICS responds directly by issuing the following messages at the console:

```
DFHTM1715 CICSITH1 CICS/ESA is being quiesced by userid IVPUSER in transaction CEMT at netname IG2S2CA8.  
DFHDM0102I applid CICS is quiescing.
```

Message DFHTM1715 is also issued to the master terminal, to inform the operator that CICS is terminating.

If the normal shutdown is successful, CICS issues the following message at the console:

```
DFHKE1799 applid TERMINATION OF CICS/ESA IS COMPLETE
```

For example, the following sequence of messages was issued on a normal shutdown of the CICS TOR, CICSHTH1:

```
15:58:41 . F CICSHTH1,CEMT PERFORM SHUTDOWN
15.05.55 . +DFHTM1715 CICSITH1 CICS/ESA is being quiesced by userid IVPUSER in
                                         transaction CEMT at netname IG2S2CA8.

15.58.42 . +DFHDM0102I CICSHTH1 CICS is quiescing.
15.58.43 . +DFHTM1782I CICSHTH1 All non-system tasks have been successfully
                                         terminated.....

15.58.44 . +DFHJC2305I CICSHTH1 Termination of VTAM sessions beginning
15.58.46 . +DFHJC2316 CICSHTH1 VTAM ACB is closed
15.58.48 . +DFHJC4553 07/15/93 15:58:48 CICSHTH1 Archive job submitted for 01 A.
15.58.49 . +DFHJC4510I CICSHTH1 All open journals now closed
15.58.52 . +DFHWK0105I CICSHTH1 Warm keypoint successful.
15.58.54 . +DFHDM0303I CICSHTH1 Transaction Dump Data set DFHDMPA closed.
15.58.55 . +DFHKE1799 CICSHTH1 TERMINATION OF CICS/ESA IS COMPLETE.
```

Shutting down CICS immediately

To shut down CICS immediately, use the CEMT PERFORM SHUTDOWN IMMEDIATE command.

What you should see

When you use the CEMT PERFORM SHUTDOWN IMMEDIATE command, CICS responds directly by issuing the DFHTM1703 message at the console. Message DFHTM1703 is also issued to the master terminal, to inform the operator that CICS is terminating.

If the CICS shutdown is successful, CICS issues the following message at the console:

```
DFHKE1799 applid TERMINATION OF CICS/ESA IS COMPLETE
```

For example, the following sequence of messages was issued on an immediate shutdown of the CICS TOR, CICSHTH1:

```
16:15:59 . F CICSHTH1,CEMT PERF SHUT IMMED
15.05.55 . +DFHTM1703 CICSITH1 CICS/ESA is being quiesced by userid IVPUSER in
                                         transaction CEMT at terminal SAMA

16.15.59 . +DFHTM1701 CICSHTH1 CICS/ESA is being terminated by operator at terminal CON1
16.16.00 . +DFHJC4553 07/15/93 15:58:48 CICSHTH1 Archive job submitted for 01 A.
16.16.00 . +DFHJC4510I CICSHTH1 All open journals now closed
16.16.01 . +DFHDM0303I CICSHTH1 Transaction Dump Data set DFHDMPA closed.
16.16.01 . +DFHKE1799 CICSHTH1 TERMINATION OF CICS/ESA IS COMPLETE.
```

Shutting down XRF CICS regions

The commands that you can use at an XRF active region to shut it down are:

- CEMT PERFORM SHUTDOWN TAKEOVER
This shuts down the active region normally, but causes it to sign off abnormally from the CAVM, and causes the XRF alternate CICS region to takeover control.
- CEMT PERFORM SHUTDOWN IMMEDIATE
This shuts down the active region immediately and causes the XRF alternate CICS region to takeover control.
- CEMT PERFORM SHUTDOWN
This shuts down both the active and alternate CICS regions.

The commands that you can use at an XRF alternate region to shut it down are:

- CEBT PERFORM SHUTDOWN

This shuts down the alternate region normally.

- CEBT PERFORM SHUTDOWN IMMEDIATE

This shuts down the alternate region normally, but causes it to sign off abnormally from the CAVM.

For more details about shutting down XRF CICS regions, see “Shutting down CICS XRF regions” on page 90.

Part 2. The CICS utility programs

This part of the book describes the CICS utility programs, and also provides some sample job streams that you can use to run them. These utility programs are as follows:

Name	Utility program use
DFHJUP	Journal select, print, and copy (see page 133)
DFHJACDU	Automatic journal archiving, for checking and updating the archiving control data set (see page 146)
DFHSTUP	Statistics formatting (see page 151)
DFHTU410	Trace formatting (see page 163)
DFHDU410	Transaction dump formatting (see page 173)
DFHDP410	SDUMP formatting appendage for use with IPCS (see page 179)
DFHMNDUP	Monitoring dictionary record creation (see page 197)
DFH\$MOLS	Sample monitoring formatting (see page 202)
DFHCSDUP	System definition data set utility (see page 215)
DFHMSCAN	Macro-level programs identification (see page 245)
DFHM\$INDB	Sample resource manager in-doubt resolution (see page 249)
DFH\$NMIG	Signon table to RACF migration (see page 253)
DFH\$STED	Sample stagger end-of-day time (see page 255)
DFHMEU	Message editing (see page 257)
DFH\$IWUP	Sample in-doubt window resolution (see page 277)

Apar PQ07674

Documentation for Apar PQ07674 added 27/10/98

DFHSMUTL Local catalog storage manager domain subpool record manipulation (see page 287)

Note: This book does not describe the IBM CICS Transaction Affinities Utility MVS/ESA, which you can use to identify possible transaction affinities that may hinder your migration to a dynamic transaction routing environment. The IBM CICS Transaction Affinities Utility MVS/ESA is available as a separate program offering, 5696-582, and is described in the *IBM CICS Transaction Affinities Utility MVS/ESA User's Guide*, SC33-1159.

Chapter 9. Journal utility programs

This chapter describes two journal utility programs:

- | | |
|-----------------|---|
| DFHJUP | The journal utility program, for reading and processing data in CICS journal data sets. |
| DFHJACDU | The journal archive control data set utility program, for checking and updating the journal archive control data set during automatic journal archiving operations. (For details of DFHJACDU, see “The automatic journal archiving utility” on page 146.) |

CICS also provides two journal utility programs for formatting tape and disk journals before they are used and another utility program for writing an end-of-file mark on tape journals:

- DFHJCJFP** used to preformat each disk data set for CICS journals, including the emergency system log data set (DFHJ01X).
- DFHFTAP** used to preformat tapes for use as CICS journals.
- DFHTEOF** used to reposition and correctly close tape journals after an uncontrolled shutdown.

For information about these utility programs, see the *CICS/ESA System Definition Guide*.

The DFHJUP utility program

You can use the journal utility program (DFHJUP) to examine data from the CICS journal data sets. The utility can:

- Print or copy an entire journal data set
- Print or copy selected journal records, from either a single or from multiple journal data sets, as specified by control statement input
- Select and print journal records on the basis of their sequential position in the data set
- Select and print journal records as determined by data contained within the records themselves, such as the contents of time, date, or identification fields
- Allow EXIT routines to process any selected journal records.

These features are selected and controlled by a series of statements that allow you to define the input and output options, selection ranges, and various field and record selection criteria.

Input and output

DFHJUP uses QSAM to process journal data that can reside on either tape or direct access storage devices. Data set organization must be physical sequential, while the record format can be fixed or variable in length, blocked or unblocked, or of undefined length. You may use multiple input and output data sets, and they may reside on different device types.

The control information must be as 80-byte records in the SYSIN data set. These control statements are reproduced on the output print data set in the same format and sequence as they are processed. If DFHJUP finds any error conditions, error messages are produced following the statement to which they apply.

You can format and print output data on the SYSPRINT data set, or copy it to a specified data set unchanged, or both.

Data to be printed is formatted into 32-byte segments and displayed in both hexadecimal and EBCDIC forms, with the hexadecimal relative offset value preceding each segment.

The flow of control for the program passes through two stages:

1. **Control statement processing**, which constructs rules for testing and selecting records, and diagnoses control statement errors.
2. **Record selection and output processing**, where the input data is read, analyzed, and compared with the selection criteria to determine the applicability of the record for output.

During the first stage, the journal utility reads and examines the parameter statements, and constructs the required test or test series to create a test group. When control passes to the next stage of the program, this test group is then used to select records. In the second stage, the input data records are read, and any action is decided by the results of each test in the group. When the end of the input data is reached, either by an end-of-file condition, or by the indicated record count being satisfied, program control shifts back to the first stage, where the next group of tests is constructed.

JCL requirements

The journal utility program runs as a standard operating system job. You must define a JOB statement, an EXEC statement, and DD statements defining input and output. "Examples of using DFHJUP" on page 141 gives some sample jobs that illustrate the use of DFHJUP.

The DD statements

STEPLIB DD

This statement defines a partitioned data set (DSORG=PO) containing the EXIT routine modules. If you are not using EXIT routines, or if the modules are in a library in the link list, this statement is not required.

SYSPRINT DD

This statement defines the output data set that will contain the formatted print records and control messages. This is usually defined as SYSOUT=A.

The DCB parameters specified for this data set are RECFM=FBA and LRECL=133. The block size may be provided on the SYSPRINT DD statement and must be a multiple of 133. The default is 133.

SYSIN DD

This statement defines the input control data set. This file must be in 80-byte record format.

Input or data DD

This statement defines the input data sets to be examined to produce the output data.

These data sets must be standard labeled files, either DASD or tape. They must be a physical sequential data sets (DSORG=PS). If a file with RECFM=U is used, the DCB BLKSIZE parameter must be specified. This only applies to records using QSAM.

Note: For CICS journal data sets, CICS builds journal records into variable length blocks before writing them, in a similar format to RECFM=VB, but with a label record in the first position of each block. To prevent accidental reblocking, journal data sets are often defined with RECFM=U; so to ensure that journal records are deblocked by DFHJUP, the DCB parameter RECFM=VB must be specified on the input data DD statement.

The default ddname is SYSUT1.

An example of a DD statement, using a variable block type of journal, is as follows:

```
//SYSUT1 DD DSNAME=CICSLOG,DISP=(OLD,KEEP),  
// DCB=RECFM=VB
```

The second example shows the use of the BLKSIZE parameter:

```
//SYSUT1 DD DSNAME=CICSLOG,DISP=(OLD,KEEP),  
// DCB=RECFM=VB,BLKSIZE=32760
```

If the journal is defined with TYPE=SMF, RECFM=U must be used.

Output or data DD

This statement defines the optional output data set(s) to contain the selected records.

DFHJUP sets the RECFM of this data set equal to the RECFM specified for the input data set. This is also done for LRECL and BLKSIZE if not specified.

The default ddname used is SYSUT4.

Utility control statements

You can use the control statements, CONTROL, OPTION, and END, to guide DFHJUP through the stages described on page 134.

Use the END statement as a delimiter to separate one group of tests (comprising one or more OPTION statements) from subsequent groups of tests on the next data set. When an END statement is encountered in the control input stream, the construction of record selection parameters ceases and the processing of input data records starts. Proper use of the END statement allows one execution of the utility program to perform a varied number of tests on one or more CICS journal data sets.

You can use the statement, * or COMMENTS, to provide titles or comments on the output listings. Use it to include any information you think is helpful to identify tests or data. It has no effect on the utility program.

Each full keyword has a corresponding abbreviated form that you may use.

You can continue keyword operands of the DFHJUP statements on the next record, up to a maximum of 9 records, provided you code a nonblank character in position

72, and continue the operands in column 16 of the next statement. If a statement is not a continuation record of the preceding statement, the character in column 72 of that preceding statement must be a blank.

CONTROL statement

The CONTROL statement (see Table 8) is optional, and you can omit it if the default operand values are satisfactory. It defines the ddnames to be used for the input and output data sets and the beginning and ending limits of the data set to be scanned. If you don't specify this statement, then DFHJUP defaults to reading the input file named in a SYSUT1 DD statement. The optional output data set defined on the SYSUT4 DD statement is opened only if you specify the OPTION COPY function in the current group of tests, and also code the COND=E parameter.

Table 8. The DFHJUP CONTROL statement		
1	10	16
CONTROL	CNTL	[,{SKIP K}={0 number}] [,{STOPAFT H}={16777215 number EOF (number,E)}] [,{DDNAME D}={SYSUT1 ddname}] [,{DDNOUT O}={SYSUT4 ddname}]

- SKIP= or K=** defines the first record tested. All prior records are ignored. If this keyword is not specified, a default value of zero is used and causes the first record on the input file to be tested.
- number** must be specified in the range 0 through 999999, and cannot have imbedded commas.
- STOPAFT= or H=** defines the last record to be tested. When this value has been reached by counting processed records, the current group of tests is terminated.
- number** must be specified in the range 0 through 9999999, with no imbedded commas.
- If you do not specify this keyword, the default value of 16777215 is assumed. If you specify a value of zero, one record is processed.
- EOF** denotes end-of-file condition; allows record processing beyond the stated maximum of 99999999 records.
- E** causes records to be counted for test sequence termination only if they satisfy selection criteria. Otherwise, all records read (after the SKIP value) are counted.
- DDNAME= or D=** identifies the ddname for the input data set for the current group of tests.
- The default ddname of SYSUT1 is used if you do not code this keyword, and a SYSUT1 DD statement must be included in your job stream. If you code this parameter to specify a different ddname, your job stream must include the corresponding DD statement.

DDNOUT= or O= identifies the ddname for the optional output data set for the current group of tests.

This keyword is used in conjunction with the OPTION COPY function, and you need only code this parameter if you want to use a ddname other than the default of SYSUT4. Coding DDNOUT, or the presence of SYSUT4 in the DFHJUP job stream, does not cause this data set to be used. An output data set is used only if OPTION COPY is specified with COND=E.

OPTION statement

The OPTION statement (see Table 9) defines the test or series of tests to be performed upon the data of the candidate record to determine whether it is selected. Each OPTION statement constructs one set of tests. You can specify one or more OPTION statements, in any combination, to define more closely the selection criteria and output processing to be performed against each input record. If you omit all keyword operands (except for EXITR and DDNAME), all records processed by stage 2 of DFHJUP are either written to the SYSPRINT data set, or copied to the specified output data set.

You can execute one or more tests on each logical record by coding the appropriate number of OPTION statements, creating the logical OR function. You can analyze records with the logical AND function by using the multifield test capability of the COND operand and the appropriate OPTION statements, creating a test series. Use the operands COND=M and COND=E to denote the beginning and ending, respectively, of a series for multifield testing of a record.

Each OPTION statement has its own output processing defaults. If you use multiple OPTION statements to create a multifield test series, final output processing is determined by the OPTION statement and its associated keywords that are defined along with the COND=E keyword.

<i>Table 9. The DFHJUP OPTION statement</i>		
1	10	16
OPTION	{PRINT	[[{OFFSET O}={1 number}]
	COPY	
	NEGOFF}	[,{FLDTYP T}={X C}]
		[,{VALUE V}=string]
		[,{FLDLEN L}={1 number}]
		[,{COND C}={E M T{Y N} ET{Y N} MT{Y N}]]
		[,{EXITR E}=name]
		[,{DDNAME D}={TRCPUNCH ddname}]
		[,{PRTSYS P}={N Y}]

Options

Each option has two distinct functions:

1. Determine the starting position for the OFFSET keyword
2. Determine the output processing to be performed.

If individual options are combined to form a multifield test, the use of OFFSET remains unchanged; however, output processing is determined by the option coded with the COND=E keyword.

PRINT

causes all selected records to be displayed on the SYSPRINT data set.

COPY

causes all selected records to be transferred to the specified output data set. You can also write these records on the SYSPRINT data set by coding the PRTSYS keyword.

NEGOF

causes the OFFSET keyword value to be used as a negative offset from the end of the journal record. All records selected using this function are displayed on the SYSPRINT data set.

All of the following OPTION control statement keywords are optional:

OFFSET= or O=

defines the location in the record of the first byte of the field to be tested. The default is position 1 of the record.

number

can be in the range from 1 up to and including the length of the record under test. Maximum value is 32767 bytes, and no checking is performed to determine if the logical record length is exceeded.

Note: If DSECTs are used to locate values in control records or blocks, you must adjust the starting value for the OFFSET parameters. Most DSECTs start with a relative value of zero, while the value specified in the OFFSET keyword is always expressed as relative to byte 1.

FLDTYP= or T=

defines the type of data in the VALUE=field.

X

data to be treated as hexadecimal pairs. The test data is packed (2 bytes into 1 to form hexadecimal equivalents). This is the default value.

Example: If VALUE=D9D6D6E3E2C5C7 (14 bytes) is specified with the FLDTYP=X parameter, the resultant VALUE= looks like this: ROOTSEG in EBCDIC characters or D9D6D6E3E2C5C7 in hexadecimal; in either case, the length is only 7 bytes.

C

data to be treated as EBCDIC characters. DFHJUP uses the data as coded in the OPTION statement, without alterations.

VALUE= or V=

defines those characters that comprise the test field. If you specify FLDTYPE=X, you must enter this data as hexadecimal character pairs. For a 'test under mask' condition, a single pair must represent the hexadecimal value for the test. If you specify FLDTYP=C, you must enter the value data as EBCDIC characters. However, if a blank or comma character is to be included in the value, you must specify FLDTYP=X, and code the value operand as hexadecimal characters, using X'40' for the blank and X'6B' for the comma, as appropriate.

string

cannot exceed 255 EBCDIC or 510 hexadecimal characters. The length of this field is set by the value of the FLDLEN= keyword and not by the number of non-null characters in this field.

FLDLEN= or L=

defines the number of characters to be used from the test field.

number

represents the actual number of bytes to be used, not the number of characters specified in the VALUE= keyword. The acceptable range of values for this field is from 1 up to and including 255. The default is 1.

COND= or C=

defines the type of test and its relationship to other tests in the group. If this keyword is not specified, the default is COND=E.

E

marks the last (or only) element in a test series. Any OPTION control statements appearing after this form a new series of tests. Coding an E to terminate a test series allows DFHJUP to perform various tests on each record, and each test series can be used on different fields within the record. Final output processing is determined by the OPTION function defined with this keyword value.

M

indicates that this is a multifield test. That is, more than one test is to be made on each input record. All tests in this series must be satisfied for record selection and output processing to begin.

T

causes the VALUE= byte to be used as a test-under-mask value, instead of as a compare field. Only the first byte (two hexadecimal characters if FLDTYP=X) of the VALUE= field is used. If FLDTYP=C is used, the hexadecimal equivalent of the EBCDIC character is the test value. If you code COND=T, you must not specify the FLDLEN= keyword and DFHJUP assumes a default length of 1.

Y

indicates that, for the test under mask to be considered satisfied, there must be a bit in the record test field for each corresponding bit of the test byte. This is equivalent to a branch-if-ones test.

N

indicates that, for the test under mask to be considered satisfied, there must not be a bit in the record test field for any of the corresponding bits of the test byte. This is equivalent to a branch-if-zeros test.

MT

defines a test-under-mask option as described above for T, but with the properties of a multifield test as described for M. Because the T parameter causes FLDLEN to default to 1, the MT parameter must be used for a multifield test that starts with a test-under-mask value.

ET

signifies that a multifield test series ends with a test-under-mask condition.

EXITR= or E=

specifies the entry point name of an exit routine that is to be given control when a candidate record has satisfied all selection criteria for the current test.

If multiple test groups have specified the same exit routine, DFHJUP attempts to load the routine into storage for each group; therefore, the routine should be reenterable. Upon reaching end of file on input, a final call is made to the exit routine. You can determine if end of file was reached by checking for zeros in the parameter field.

The interface to the exit routine is as follows:

ENTRY:**REGISTERS**

R1 contains a pointer to a parameter list.
R13 points to an empty save area.
R14 contains a return address.
R15 contains the exit routine entry address.

PARMLIST

The parameter list consists of 2 words. The first is a pointer to the candidate record; the second (with the high order bit on) is a pointer to the SYSPRINT data set DCB.

EXIT:

Upon return from the exit routine, the contents of register 15 determine whether or not processing is to continue on this record.

A nonzero value indicates that no further processing is to be done on this record, and selection tests start again against the next input record.

A zero value indicates that this record is required, and output processing is now determined by the last OPTION statement encountered containing the COND=E keyword.

If the EXITR keyword is omitted, processing continues as if a return code value of zero was received.

DDNAME= or D=

defines the output data set used by the DL/I call trace journal record retrieval routine for whenever it has been specified as the user exit routine. A corresponding DD statement must be supplied.

PRTSYS= or P=

determines whether to print all the selected records on the SYSPRINT data set.

N

indicates that no printing of selected records is to be done.

Y

indicates that all records transferred to the output data set are also formatted and printed.

This keyword can be used only with the OPTION COPY function. N is the default.

END statement

When you have defined all tests for the current input file, use the END statement (see Table 10) to initiate the tests.

Positions 10 and upward can be used for comments.

1	10	16
END	[.....comments.....]	

COMMENTS statement

The COMMENTS statement (see Table 11) is optional. If used, it causes the contents to be displayed on the SYSPRINT data set.

1	10	16
*		

DFHJUP return codes

The following errors can occur in DFHJUP:

<i>Return code</i>	<i>Error description</i>
--------------------	--------------------------

- | | |
|-----------|---|
| 04 | This return code can be issued for any of the following reasons: <ul style="list-style-type: none">• A syntax error is detected in a SYSIN record• A syntax error is detected in an OPTION statement• A SYSIN statement type is unknown• An OPEN failure has occurred on SYSUT1 or SYSUT4. |
| 08 | An I/O error has occurred on either SYSUT1 or SYSUT4. |
| 16 | Either SYSIN or SYSPRINT has failed to open. |

Examples of using DFHJUP

The following examples illustrate some of the ways in which DFHJUP can be used. Each makes reference to a CICS journal data set. However, this utility can be used with any data set that can be processed using QSAM.

For clarity, all option keywords have been specified in their full form, and many are coded where the default could be taken. Use of the short form and keyword defaults will greatly reduce the required input. In each of the two main examples, the COMMENT statement has been used to describe the function being performed.

Example 1

Figure 30 on page 142 shows the JCL and control statements required to print all the records from a CICS journal data set, which is a DASD file, to the output data set.

Examples of the use of the OPTION parameters

The OPTION parameters can be used to select specific types of records from a journal. A table has been provided at the end of this section to assist in the construction of these statements. The following passages contain code fragments which illustrate some of the types of record selection that can be achieved in this way.

Locating records using the system-type ID field: This example shows how to locate all records in a journal which are generated by the syncpoint program and mark the start of a unit of work. The two byte hexadecimal number, used with the VALUE parameter, is found from the tables under *CICS Journaling* in the *CICS/ESA Customization Guide*. The module identifier is located first, (in this case it is X'59'), and this is then used to find the relevant section in the function identifiers table. (Our example uses X'EF'). In the journal, the system-type ID field is made up of the function identifier followed by the module identifier. The offset to the start of this field is 5, and it is a numeric (X) type of field, of length 2 bytes. The value we are going to use is X'EF59', as shown in the following code fragment:

```
//SYSIN      DD *
OPTION PRINT OFFSET=5,FLDTYP=X,VALUE=EF59,FLDLEN=2,COND=E
END
/*
```

If all the file control records were to be found, for example, then only the second half of the system-type ID would be used. The OPTION statement would have the following form:

```
//SYSIN      DD *
OPTION PRINT OFFSET=6,FLDTYP=X,VALUE=11,FLDLEN=1,COND=E
END
/*
```

Using the task number: The task number appears as a three byte packed decimal value in a journal record. It must appear in the same form in the VALUE parameter. To do this take the actual task number, in this case 25, and turn it into a five digit decimal value by filling up the left hand side with zeros: 00025. Then add a capital letter C to the right hand end to show its a positive value: 00025C. The following statements will cause all records belonging to task 25 to be directed to the SYSPRINT data set:

```
//SYSIN      DD *
OPTION PRINT OFFSET=16,FLDTYP=X,VALUE=00025C,FLDLEN=3,COND=E
END
/*
```

Finding all records for a particular transaction: The transaction identifier appears as a 4-byte hexadecimal field in the journal records. If FLDTYP=C is used in the parameters then this value can be entered in its character form as shown below, for a transaction called TRN1.

```
//SYSIN      DD *
OPTION PRINT OFFSET=23,FLDTYP=C,VALUE=TRN1,FLDLEN=4,COND=E
END
/*
```

Alternatively, the hexadecimal equivalent for these characters could be used, with FLDTYP=X, as shown in the next example.

```
//SYSIN      DD *  
OPTION PRINT OFFSET=23,FLDTYP=X,VALUE=E7F0F0F5,FLDLEN=4,COND=E  
END  
/*
```

Finding all records with a particular time stamp: Here the time must be entered in the form hhhmsss+ as a series of decimal digits and where the + sign is represented by the letter F. The utility does not support the use of the 'greater than' or 'less than' logical operators, so searching using a time stamp value is of limited use.

```
//SYSIN      DD *  
OPTION PRINT OFFSET=19,FLDTYP=X,VALUE=1446591F,FLDLEN=4,COND=E  
END  
/*
```

Locating all records from a particular terminal: The terminal identifier is a 4-byte value which can be entered as four characters or their hexadecimal equivalent, in the same way as a transaction identifier. In this example all the records from terminal T004 are to be selected and printed.

```
//SYSIN      DD *  
OPTION PRINT OFFSET=27,FLDTYP=C,VALUE=T004,FLDLEN=4,COND=E  
END  
/*
```

Selection using more than one search parameter: Suppose you wanted to print all the start of unit of work records for a particular task. This needs two OPTION statements. The example shows how to search for all records which belong to task number 25 and have a system type ID of X'EF59'. The COND=M parameter performs the AND operation on the two statements.

```
//SYSIN      DD *  
OPTION PRINT OFFSET=16,FLDTYP=X,VALUE=00025C,FLDLEN=3,COND=M  
OPTION PRINT OFFSET=5,FLDTYP=X,VALUE=EF59,FLDLEN=2,COND=E  
END  
/*
```

If more than one type of record is to be found then the form of the following example could be used. In this case all the file control records for task 48 are selected together with all the records generated by the TRN6 transaction. The first two OPTION statements are 'ANDed' together, whereas the third statement is a separate search because the second statement is terminated by COND=E.

```
//SYSIN      DD *  
OPTION COPY OFFSET=6,FLDTYP=X,VALUE=11,FLDLEN=1,COND=M  
OPTION COPY OFFSET=16,FLDTYP=X,VALUE=00048C,FLDLEN=3,COND=E  
OPTION COPY OFFSET=23,FLDTYP=C,VALUE=TRN6,FLDLEN=4,COND=E  
END  
/*
```

Using a mask to test specific bits in a record: Individual bit values may be used to select specific records from a journal in much the same way as the previous examples. The OFFSET and VALUE must be known. In this example the first option statement tests for all file control records. This is combined with the second statement to test for those records which were written by emergency

restart. The OFFSET is 13, to the start of the JCSPFS field, plus 2 to move to the last byte in the field. You can find the value using the programming information in the *CICS/ESA Customization Guide*. The length field must not be included. COND=ET shows it is a test under mask and the end of a multifield test series.

```
//SYSIN      DD *
OPTION COPY OFFSET=6,FLDTYP=X,VALUE=11,FLDLEN=1,COND=M
OPTION COPY OFFSET=15,FLDTYP=X,VALUE=20,COND=ET
END
/*
```

OPTION parameter values

<i>Table 12 (Page 1 of 2). OPTION parameter values</i>					
Field name	OFFSET	FLDTYP	VALUE (example)	FLDLEN	Contents
Note: System header fields					
JCRLL	1	X	0037	2	Length of record
JCRBB	3	X	0000	2	Not used
JCRSTRID	5	X	EF59	2	System type ID
	5	X	EF	1	Function identifier
	6	X	59	1	Module identifier
JCRUTRID	7	X	12EF	2	User type ID
JCRLRN	9	X	002C	2	Record number within block
Note: Main system prefix fields					
JCSPLL	11	X	0014	2	Length of system prefix
JCSPFS	13	X	000002	3	Flags
JCSPTASK	16	X	00025C	3	Task number
JCSPTIME	19	X	1445123F	4	Time of request - hhmmsss+
JCSPTRAN	23	C	TRN1	4	Transaction identifier
	23	X	E3D9D5F1	4	alternative format

Table 12 (Page 2 of 2). OPTION parameter values					
Field name	OFFSET	FLDTYP	VALUE (example)	FLDLEN	Contents
JCSPTERM	27	C	T004	4	Terminal identifier
	27	X	E3F0F0F4	4	alter-native format

The automatic journal archiving utility

This section describes the function of the DFHJACDU utility program, which is used in conjunction with the automatic journal archiving option.

The DFHJACDU utility program

CICS maintains a journal archive control data set (JACD) for all CICS disk journals that have JOUROPT=AUTOARCH defined in their JCT entries. You can use the JACD utility program, DFHJACDU, to read or update the JACD. There are three DFHJACDU commands, which you can use to:

- **Check** the status of a specified journal at the start of a journal archive job
- **Update** the status of a specified journal at the end of a journal archive job
- **List** the journal archive control records from the JACD.

You would normally use the first two of these commands, CHECK and UPDATE, within an automatic journal archive job that is submitted by the journal archive submission program, DFHJASP. You should write automatic archive jobs in the form of skeletal JCL, in which you code the CHECK and UPDATE commands with symbolic parameters, leaving DFHJASP to substitute the required values. For details of the symbolic parameters that are supported by the archive submission program, see “Symbolic parameter substitution” on page 148.

You can also code the CHECK and UPDATE parameters manually to submit an archive job, but it is unlikely that you would want, or need, to do so.

You do not use symbolic parameters with the LIST command, which you code in a batch job for submission in the usual way.

The DFHJACDU commands

You specify the required function by coding the DFHJACDU command in a PARM parameter on the EXEC PGM=DFHJACDU statement. The format of each command and its parameters is described in the following sections.

The DFHJACDU CHECK and UPDATE commands

When you code the CHECK command in an automatic journal archive job, DFHJACDU reads the control record for the specified journal and checks the status. This command should be coded in the first step of an archive job, to ensure that the journal data set is in the correct state to be archived. DFHJACDU sets the return code to indicate the journal status as follows:

Return code	Description
0	NOTREADY, and the DATE and TIME parameters agree with the JACD data
4	NOTREADY, and the DATE and TIME parameters do not agree with the JACD data
8	READY
12	CURRENT
16	JACD I/O error.

The journal archive step should only run if the return code is zero.

When you code the UPDATE command, DFHJACDU again checks the status to make sure that it is still NOTREADY, and if this is the case, changes the status to READY. In an automatic journal archive job, you must code the COND parameter on the DFHJACDU EXEC statements, to ensure that job steps only execute if the return code from the previous step is zero.

The full syntax of the commands is as follows:

CHECK|UPDATE

JOURNAL=jfileid,DATASET=A|B|X[,DATE=yyyymmdd][,TIME=hhmmss]

Code the CHECK command to test the status of the specified journal data set. You specify which journal data set by coding the JOURNAL and DATASET parameters. You can optionally add the DATE and TIME parameters as an extra check where more than one archive job may be submitted for a given journal data set.

Code the UPDATE command to change the status of the specified journal data set, after successful archiving, from NOTREADY to READY.

JOURNAL In your skeletal JCL, code the JOURNAL operand with the symbolic parameter %JJ. DFHJASP substitutes the appropriate journal identifier in the range 01 through 99, where 01 is the system log and 02 through 99 are user journals.

DATASET In your skeletal JCL, code the DATASET operand with the symbolic parameter %D. DFHJASP substitutes the appropriate data set identifier, which can be A, B, or X (X is only applicable for the system log).

DATE In your skeletal JCL you can, optionally, code this with %ODATE, and DFHJASP substitutes the date the journal data set was opened. This provides additional identification of the journal data set.

TIME In your skeletal JCL you can, optionally, code this with %OTIME, and DFHJASP substitutes the time the journal data set was opened. This provides additional identification of the journal data set.

The DFHJACDU LIST command

LIST [JOURNAL=(jfileid1,jfileid2,....,jfileidn)]

Code the LIST command to request a formatted listing of the specified journal control records. You identify journals by the journal identifier (JFILEID in the JCT), where 01 is the CICS system log, and 02 through 99 are user journals. If you omit the JOURNAL parameter, DFHJACDU lists the entire JACD.

Symbolic parameter substitution

For an automatic journal archiving job, symbolic parameter substitution is available on all of the parameters. The following are the symbolic parameters that you can use with journal archive job submission:

Symbolic Parameter	Represents
%SYST	A four-byte identifier of the CICS region issuing the journal archive job, derived from the SYSIDNT system initialization parameter.
%JJ	A two-digit journal identification (JFILEID), in the range 01 through 99.
%D	A 1-character dataset identification: A, B, or X.
%JOURDSN	A 1- to 44-byte journal dataset name.
%ODATE	A 7-byte journal dataset open date (yyyymmdd).
%OTIME	A 7-byte journal dataset open time (hhmmss).
%CDATE	A 7-byte journal dataset close date (yyyymmdd).
%CTIME	A 7-byte journal dataset close time (hhmmss).
%JACDDSN	A 1- to 44-byte JACD dataset name.
%APPLID	A 1- to 8-byte CICS APPLID. Note that for an XRF system this is the generic, not the specific, APPLID.

The journal archive submission program carries out substitution on all symbolic parameters it finds in the JCL statements in the skeletal JCL, except JCL comment statements. The program copies JCL comment statements and non-JCL statements as they are. It also copies positions 72-80 of JCL statements without any changes. If the substitution causes the statement to exceed position 71, then DFHJASP generates a continuation statement, continuations starting in position 5 (after the //). Positions 72-80 of a continuation are kept the same as in the skeletal statement being continued.

Sample DFHJACDU job stream

The job shown in Figure 32 on page 149 illustrates the use of symbolic parameter substitution in a journal archive skeletal job. There are some notes following the sample job explaining some of the job control statements.

```

//$SYST%JJ%D JOB accounting info,'
//      CLASS=A,MSGCLASS=A,NOTIFY=userid
//CHECK EXEC PGM=DFHJACDU,
//      PARM='CHECK JOURNAL=%JJ,DATASET=%D,DATE=%ODATE,TIME=%OTIME'
//STEPLIB DD DSN=CICS410.SDFHLOAD,DISP=SHR
//DFHJACD DD DSN=%JACDDSN,DISP=SHR
//SYSPRINT DD SYSOUT=*
//DFHJAPRT DD SYSOUT=*
//*
//ARCHIVE EXEC PGM=IEBGENER,COND=(0,NE)
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
//SYSUT1 DD DSN=%JOURDSN,DISP=SHR,
//      DCB=RECFM=VB
//SYSUT2 DD DSN=CICS410.ARCHIVE.J%JJ.D%CDATE.T%CTIME,
//      DISP=(NEW,CATLG,DELETE),
//      DCB=*.SYSUT1,VOL=SER=volid,UNIT=TAPE
//*
//UPDATE EXEC PGM=DFHJACDU,COND=(0,NE),
//      PARM='UPDATE JOURNAL=%JJ,DATASET=%D,DATE=%ODATE,TIME=%OTIME'
//STEPLIB DD DSN=CICS410.SDFHLOAD,DISP=SHR
//DFHJACD DD DSN=%JACDDSN,DISP=SHR
//SYSPRINT DD SYSOUT=*
//DFHJAPRT DD SYSOUT=*
//*
//

```

Figure 32. Sample DFHJACDU job to check and update the JACD

Notes:

1 The PARM statement on this EXEC DFHJACDU statement is coded with the CHECK command, to test the status of the specified journal data set. The command is coded with symbolic parameters, which are substituted by DFHJASP at job submission time.

2 The archive job step is coded with the COND parameter, to test the return code from the first job step. This ensures that the journal is only archived if the:

- Status of the journal data set is NOTREADY
- Open date in the CHECK parameter agrees with the open date in the JACD
- Open time in the CHECK parameter agrees with the open time in the JACD.

If you are running CICS with local DL/I support, you should use the DL/I archive utility, DFSUARCO, to copy journal data sets.

3 The third job step is coded to update the status of the specified journal data set from NOTREADY to READY. The COND parameter is coded to ensure that this step executes only if the return code from the archive job step is zero.

Chapter 10. Statistics utility program, DFHSTUP

The statistics utility program, DFHSTUP, prepares and prints reports offline, using the CICS statistics data recorded on the MVS system management facilities (SMF) SYS1.MANx data sets.

PN65335

The following change was made by APAR PN65335.

To enable the CICS statistics domain to record interval statistics on these SMF data sets, you must specify the STATRCD=ON system initialization parameter. The other statistics record types (unsolicited, requested and end-of-day) are written regardless of the setting of the STATRCD option. For information about the SMF data sets, see the *MVS/ESA System Management Facilities* manual. For information about what CICS data is recorded on the SMF data sets, and about interpreting CICS statistics output in the DFHSTUP report, see the *CICS/ESA Performance Guide*. For a description of the STATRCD system initialization parameter, see the *CICS/ESA System Definition Guide*.

Use the version of the DFHSTUP program from the same release of CICS as the data that it is to process. This chapter describes the CICS/ESA 4.1 version of the DFHSTUP program, which you should use for CICS/ESA 4.1 data **only**. If you want to prepare and print reports offline of data from another release of CICS, use the version of the DFHSTUP program for that release.

The statistics recording status

The statistics recording status is set at CICS startup by the system initialization parameter STATRCD.

STATRCD	Meaning
---------	---------

OFF (default)	Interval statistics are not collected.
----------------------	--

	End-of-day, Unsolicited, and Requested statistics are written to SMF regardless of the STATRCD setting.
--	---

ON	Interval statistics are collected.
-----------	------------------------------------

	On a cold start of a CICS region, interval statistics are recorded by default at three-hourly intervals. All intervals are timed using the end-of-day time (midnight is the default) as a base starting time (not CICS startup time). This means that the default settings give collections at 00.00, 03.00, 06.00, 09.00, and so on, regardless of the time that you start CICS.
--	---

PN65335

The following change was made by APAR PN65335.

You can use the global user exit XSTOUT in the statistics domain, to give control to an exit program before each statistics record is written to the SMF data set, and

+ can access the relevant statistics record. You may want to use an exit program at
+ this exit point to examine the statistics record and suppress the writing of unwanted
+ records. For information about the XSTOUT global user exit, see the *CICS*
+ *Customization Guide*.

On a warm or emergency restart the statistics recording status is restored from the CICS global catalog.

You can change the statistics recording status at any time, for example as follows:

- During a warm or emergency restart by changing the STATRCD system initialization parameter as a SIT override.
- While CICS is running by using the CEMT or EXEC CICS SET STATISTICS command.

Whatever the value of the STATRCD system initialization parameter, you can ask for requested statistics and requested reset statistics to be collected. You can get statistics “on demand” for all, or for specified, resource types by using the CEMT or EXEC CICS PERFORM STATISTICS command. The period covered for statistics requested in this way is from the last reset time up to the time that you issue the PERFORM STATISTICS command.

The last reset time is either of the following:

- The beginning of the current interval
- The logical end-of-day collection time
- The time that you last issued a CEMT or EXEC CICS SET or PERFORM STATISTICS command specifying RESETNOW.

For details of how to use the CEMT PERFORM STATISTICS and SET STATISTICS commands, see the *CICS/ESA CICS-Supplied Transactions* manual. For programming information about the equivalent EXEC CICS commands, see the *CICS/ESA System Programming Reference* manual.

Whenever you use a CEMT or EXEC CICS SET command to change the statistics recording status, the changed status is recorded in the global catalog for use in a warm or emergency restart.

“Job to run the DFHSTUP program” on page 153 gives information about how to use the DFHSTUP program to select and format CICS statistics.

Restriction in the use of some output devices

Statistics data is written by CICS to the SMF data sets in a mixture of upper and lower case English characters. By default, the DFHSTUP program outputs the data in a mixture of upper and lowercase characters. If the keyword UPPERCASE=YES is coded in the SYSIN data stream, all data is output in uppercase only. This provides support for Katakana devices.

Job to run the DFHSTUP program

The job shown in Figure 33 comprises three job steps, of which the second step is optional. The job steps are:

1. Unload the SMF data set (or data sets) containing the CICS statistics that you want to process.
2. Optionally, run the journal utility program to extract from the dumped SMF data set(s) the statistics record types only. If you are interested in the statistics for a single CICS region only, you can add another OPTION statement to select the statistics records for the CICS APPLID you want.

Note: Using the journal utility to select APPLIDs is optional; the statistics utility program also allows you to select by specific APPLID. In this example we are selecting records using the generic APPLID at offset 47. If you want to select records for the specific APPLID of a CICS region running with XRF, you should specify OFFSET=55.

3. Run the statistics utility program to sort, format, and print the statistics data.

```
//STUP      JOB accounting info,CLASS=A,
//          USER=userid,MSGCLASS=A,NOTIFY=userid
//*****
/* Step 1: Unload data from the SMF data sets
//*****
//SMFDUMP  EXEC PGM=IFASMFDP
//INDD1    DD DSN=SYS1.MANx,DISP=SHR,AMP=('BUFSP=65536')           1
//INDD2    DD DSN=SYS1.MANy,DISP=SHR
//OUTDD1   DD DSN=user.SMF.DATA,DISP=(NEW,CATLG),                 2
//          SPACE=(CYL,(50,10)),UNIT=SYSDA
//SYSPRINT DD SYSOUT=A
//SYSIN    DD *
           INDD(INDD1,OPTIONS(DUMP))                               1
           INDD(INDD2,OPTIONS(DUMP))
           OUTDD(OUTDD1,TYPE(0:255))                               2

/*
//*****
/* Step 2: Optionally, select and copy the statistics records only
/*          to a journal-type data set using the journal utility
//*****
//JUP EXEC PGM=DFHJUP
//STEPLIB DD DSN=CICS410.SDFHLOAD,DISP=SHR
//SYSUT1  DD DSN=user.SMF.DATA,DISP=SHR
//SYSUT4  DD DSN=&&SMFSTATS,DISP=(NEW,PASS),UNIT=SYSDA,
//          SPACE=(CYL,(30,10))
//SYSPRINT DD SYSOUT=A
//SYSIN   DD *
OPTION COPY OFFSET=6,FLDTYP=X,VALUE=6E,FLDLEN=1,COND=M           3
OPTION COPY OFFSET=23,FLDTYP=X,VALUE=0002,FLDLEN=2,COND=M
OPTION COPY OFFSET=47,FLDTYP=C,VALUE=genapplid1,FLDLEN=8,COND=E
* The next test group is only needed to select a second APPLID
OPTION COPY OFFSET=6,FLDTYP=X,VALUE=6E,FLDLEN=1,COND=M           3
OPTION COPY OFFSET=23,FLDTYP=X,VALUE=0002,FLDLEN=2,COND=M
OPTION COPY OFFSET=47,FLDTYP=C,VALUE=genapplid2,FLDLEN=7,COND=E
/*
//*****
/* Step 3: Sort, format and print the statistics records          4
//*****
//STUP1   EXEC PGM=DFHSTUP,REGION=32M                             5
//*****
//STEPLIB DD DSN=CICS410.SDFHLOAD,DISP=SHR
//          DD DSN=CICS410.SDFHAUTH,DISP=SHR
```

Figure 33 (Part 1 of 2). Example job to extract and print statistics data

```

//DFHSTATS DD DSN=&&SMFSTATS,DISP=(OLD,DELETE)
//*DFHSTATS DD DSN=user.SMF.DATA,DISP=SHR
//DFHSTWRK DD UNIT=SYSDA,SPACE=(CYL,(8,4))
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(4))
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(4))
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(4))
//SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,(4))
//SORTWK05 DD UNIT=SYSDA,SPACE=(CYL,(4))
//DFHPRINT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SYSIN DD *
SELECT APPLID=(applid1,applid2)
COLLECTION TYPE=ALL
/*
//

```

Figure 33 (Part 2 of 2). Example job to extract and print statistics data

Notes:

1 You can specify any number of input (INDD) and output (OUTDD) data sets for the SMF dump program, IFASMFDP. The input files are dumped in reverse order unless concatenated under one input file. For example, in Figure 33 on page 153, two input files are specified. After the IFASMFDP program is processed, the output file (user.SMF.DATA) contains the records from INDD2 first, followed by the records for INDD1. Although you probably code the INDD parameter and the associated DD statements to process the data sets in chronological order, the DFHSTUP program produces a correct report if you fail to do so.

For further information about unloading SMF data sets, see the *MVS/ESA System Management Facilities* manual.

Note: The AMP parameter is used on the DD statement to reduce the unload time if you specify a suitable buffer size. (See also the monitoring utility sample job on page 201.)

2 The OUTDD parameter refers to an OUTDD1 DD statement defining a disk data set. To rerun this job you need to change the DISP parameter to DISP=(OLD,KEEP). The TYPE parameter specifies the full range of SMF record types, causing IFASMFDP to unload all records. If you want to select only CICS records, change the record type parameter to TYPE(110).

3 The OPTION statements define which records DFHJUP copies to the output data set defined by the SYSUT4 DD statement. In this example, they comprise two sets of multifield tests designed to select only the CICS statistics records for two APPLIDs. (The selection option shown here is selecting by generic APPLID.) The number of test groups you code depends on the number of CICS APPLIDs you want to select from the SMF data. The tests are specified as:

Test Group 1 The first three OPTION statements select record type 110 (HEX value 6E), but only those that are sub-type 0002 (statistics), for the CICS region with the generic applid *genapplid1*.

Test Group 2 The last three OPTION statements also select record type 110 (HEX value 6E), and those that are sub-type 0002 (statistics), but for the CICS region with the generic applid *genapplid2*.

You cannot specify blanks in a value defined as FLDTYP=C, so if your APPLID is less than the maximum of eight characters, you must not code trailing blanks, and code the FLDLEN operand with the exact length.

Using DFHJUP OPTION parameters to select APPLIDs is optional, as you can also select APPLIDs using parameters of the DFHSTUP program after the sort step. However, selecting at this stage can reduce the amount of data to be sorted if you know that you only want to print some APPLIDs. If you omit the third OPTION parameter in each test group, you must change the COND=M parameter on the second OPTION statement to COND=E.

For more information about coding DFHJUP OPTION parameters, see “OPTION statement” on page 137.

4 The DFHSTUP program sorts statistics records in the sequence: specific applid, date (in DDYY form), and time.

5 Unless there are storage constraints under which the DFHSTUP program must run, we recommend that you use a region size of 32MB. This gives the best performance and ensures that the DFHSTUP program is allocated all available storage both below and above the 16MB line.

If storage is limited, the region size may be reduced at the cost of reducing performance. We recommend that you use a region size of 820KB for all SMF data sets (listed in the DFHSTATS DD concatenation) up to about 1MB in size, provided that you specify at least one cylinder of disk storage via the SORTWKnn data sets. For SMF data sets greater than 1MB, you may need to increase the region size; increment the region size in 20KB steps (For example, 840KB, 860KB, 880KB, and so on.) for maximum efficiency. Performance may be improved by increasing the disk space allocated to the SORTWKnn data sets.

6 The ddname for the input to the DFHSTUP program must be DFHSTATS.

If you have not used step 2 to copy the CICS statistics records to the &&SMFSTATS data set, change the DD statement to specify the user SMF data set, as follows:

```
//DFHSTATS DD DSN=user.SMF.DATA,DISP=SHR
```

7 The ddname for the DFHSTUP work file must be DFHSTWRK. The following calculation can be used to estimate the size of the DFHSTWRK data set required:

1. The length of the file statistics dsect, DFHA17DS, is 304 bytes.
2. The amount of file data written in *one* interval/requested reset/End-of-day report is $304 * (\text{number of files in region})$ bytes.
3. Additionally, as file statistics are written to SMF whenever a file is closed online, an estimate of the number of file closes, multiplied by the length of the DFHA17DS dsect (304 bytes), should be added.

8 The DFHSTUP program sorts the data by means of a link to the MVS sort program, DFSORT, to ensure that data is correctly processed in chronological sequence. These sort work files are needed by the DFSORT program.

9 The ddname for the output from the DFHSTUP program must be DFHPRINT, which you can direct either to a data set or printer.

10 The control parameters for the DFHSTUP program can be supplied only in the SYSIN data set.

Each control parameter in the SYSIN data set should start on a new line and is terminated by a blank. If you need to continue a control parameter for more than one line, you must ensure that the line to be continued ends with a comma in column 1 through 71, there is a non-blank character in column 72 of the line to be continued, and start each continuation line in column 16. For example:

```
/* Column 16 v                                     72 v
SELECT APPLID=(CICSIDA,CICSIDB,CICSIDC,CICSIDD,CICSIDE,CICSIDF,CICSIDG,/
              CICSIDH,CICSIDI,CICSIDJ)
COLLECTION TYPE=ALL
```

The available parameters are described in “Control parameters of the DFHSTUP program.”

Control parameters of the DFHSTUP program

This section describes the parameters that you can use to control the output of statistics data by the DFHSTUP program.

Note: If you want the statistics output in uppercase only, you must code the UPPERCASE=YES parameter first in the parameter list.

If you do not code any parameter, the DFHSTUP program formats all the collection types for all APPLIDs, to a page size of 60 lines.

SELECT APPLID={applid}(applid1[,applid2]..[,applidN])

Code this parameter with the applids of the CICS regions for which you want statistics to be formatted and printed. The parameter keywords must be coded as shown, with one blank between the two words. Code only one SELECT APPLID parameter or one IGNORE APPLID parameter, with up to 40 APPLIDs. If you specify more than 40 APPLIDs, the results are unpredictable.

If your CICS regions are defined with both generic and specific APPLIDs, it is the specific APPLID that you must specify on the SELECT APPLID parameter.

If you do not code this parameter, the DFHSTUP program reports statistics for all APPLIDs found in the DFHSTATS data set, other than those APPLIDs specified on an IGNORE APPLID parameter.

IGNORE APPLID={applid}(applid1[,applid2]..[,applidN])

Code this parameter with the APPLIDs of the CICS regions for which you want the statistics ignored. The parameter keywords must be coded as shown, with one blank between the two keywords. Code only one SELECT APPLID parameter or one IGNORE APPLID parameter, with up to 40 APPLIDs. If you specify two or more APPLIDs, you must enclose them in parentheses, and separate them by commas. If you specify more than 40 APPLIDs, the results are unpredictable.

If your CICS regions are defined with both generic and specific APPLIDs, it is the specific APPLID that you must specify on the IGNORE APPLID parameter.

If you do not code this parameter, the DFHSTUP program reports statistics for all APPLIDs found in the DFHSTATS data set, according to the SELECT APPLID parameter.

The parameter keywords must be coded as shown, with one blank between the two words. The operands are as follows:

- ALL** Format the statistics for all types of collection, for all the specified APPLIDs: this is the default.
- INT** Format the statistics that were collected at specified intervals, for all of the selected APPLIDs.
- EOD** Format end-of-day type statistics for all of the selected APPLIDs. The end-of-day collection type is simply another interval collection, being the final collection of statistics for the last interval at the time of shut down or logical end-of-day. Because the specified interval over which interval statistics are being collected may not have elapsed, the actual interval spanning the end-of-day collection can be a short interval.
- REQ** Format requested statistics for all of the selected APPLIDs.
- RRT** Format requested reset statistics for all of the selected APPLIDs. These are statistics asked for by using the CEMT or EXEC CICS PERFORM STATISTICS RESETNOW command or, when changing the statistics recording status, the CEMT or EXEC CICS SET STATISTICS ON|OFF RECORDNOW RESETNOW command.
- USS** Format unsolicited statistics for all of the selected APPLIDs. CICS collects unsolicited statistics for:
 - Autoinstall** Whenever an autoinstalled terminal entry in the TCT is deleted (after the terminal logs off), CICS collects statistics covering the autoinstalled period since the last interval. This period covers any delay interval specified on the SIT parameter AILDELAY.

If an autoinstall terminal logs on again before the expiry of the delay interval, then the accumulation of statistics continues until the next interval. At that interval, the accumulation of statistics is restarted.
 - DBCTL** Whenever CICS disconnects from DBCTL, CICS collects the statistics covering the whole of the DBCTL connection period.
 - FEPI connections** Unsolicited connection statistics are produced when a connection is destroyed. This could occur with a DISCARD TARGET, DISCARD NODE, DISCARD POOL, DELETE POOL, DISCARD NODELIST, or DISCARD TARGETLIST command is used.
 - FEPI pools** Unsolicited Pool statistics are produced when a pool is discarded (DISCARD POOL or DELETE POOL).
 - FEPI targets** Unsolicited target statistics are produced when a target is destroyed or removed from a pool. This occurs when a DELETE POOL, DISCARD POOL, DISCARD TARGET, or DISCARD TARGETLIST command is used.
 - Files** Whenever CICS closes a file, CICS collects statistics covering the period from the last interval.

LSRPOOL When CICS closes the last file in an LSRPOOL, CICS collects the statistics for the LSRPOOL. The following peak values are reset to the current value at each interval collection:

- Peak number of requests waiting for a string
- Maximum number of concurrent active file control strings.

The other statistics, which are not reset at an interval collection, cover the entire period from the time the LSRPOOL is created (when the first file is opened) until the LSRPOOL is deleted (when the last file is closed).

Programs Whenever an installed program definition is discarded, CICS collects the statistics covering the installed period since the last interval.

Transactions Whenever an installed transaction definition is discarded, CICS collects the statistics covering the installed period since the last interval.

Transaction classes Whenever an installed transaction class definition is discarded, CICS collects the statistics covering the installed period since the last interval.

TIME START=hh.mm.ss,STOP=hh.mm.ss

Code this to cause the DFHSTUP program to print only the statistics collected during the specified period (START time to STOP time). This parameter should be used in conjunction with the DATE parameter. If no DATE parameter is coded, statistics collected during the specified period are printed regardless of the date on which they were collected. The parameter keywords must be coded exactly as shown, with one blank between the first two words, and with both START and STOP times specified. The start and stop times must be specified as:

hh.mm.ss where:

hh = number of hours (24 hour clock notation)

mm = number of minutes

ss = number of seconds.

Note: If the specified period (START time to STOP time) spans across midnight, the DATE parameter must also be coded.

+ **DATE START=mm/dd/yy or mm/dd/yyyy,STOP=mm/dd/yy or mm/dd/yyyy**
Code this to force the DFHSTUP program to print only statistics collected during the specified period (START date to STOP date). This parameter should be used in conjunction with the TIME parameter. If no TIME parameter is coded, statistics collected at any time during the specified period are printed. The parameter keywords must be coded exactly as shown, with one blank between the first two words, and with both START and STOP dates specified. The start and stop dates must be specified as:

+ **mm/dd/yy or mm/dd/yyyy** where:
+ mm = month of the year
+ dd = day of the month
+ yy = year of the twentieth century
+ yyyy = year
+ yyyy must be coded to represent a date in the twenty-first century. If
+ yy is coded the twentieth century is assumed. For example, a date of
+ 12/20/95 is for the 20th December 1995 and a data of 12/20/2005 is
+ for the 20th December 2005.

PAGE SIZE=number
Code this with the number of lines to be formatted per page, in the range 20 to 99. The default page size is 60 lines.

SUMMARY
Code this if you require the DFHSTUP program to produce a summary report for each APPLID selected. A summary report is composed by adding together the statistics contained in the interval, requested reset, end-of-day, and unsolicited collections. The summary report statistics are listed in almost the same order as interval and end-of-day reports. The only difference is that DBCTL statistics appear at the end of the summary. DBCTL statistics are unsolicited only, so you do not get them for interval, requested reset, or end-of-day collections.

The summary report lists statistics records in the following type order:

- | Statistics domain
- | Transaction manager
- | Transaction class
- | Dispatcher
- | Monitoring
- | Storage Manager DSA
- | Storage Manager task subpool
- | Storage Manager domain subpool
- | Loader
- | Temporary storage
- | Transient data
- | VTAM
- | Terminal Autoinstall
- | Program Autoinstall
- | System dump
- | Transaction dump
- | Dynamic transaction backout
- | Table manager
- | Transaction
- | Program

Chapter 11. Trace utility print programs

There are three destinations for CICS region trace data:

1. A table in main storage, when you specify INTTR=ON and SYSTR=ON as system initialization parameters
2. The CICS auxiliary trace data sets, when you specify AUXTR=ON and SYSTR=ON as system initialization parameters
3. The MVS/ESA generalized trace facility (GTF) data sets, when you specify GTFTR=ON and SYSTR=ON as system initialization parameters.

You can also obtain trace entries at these destinations while CICS is running, by means of the CETR trace transaction or the equivalent EXEC CICS SET commands.

This chapter describes how you can print the CICS region trace data from:

- The CICS auxiliary trace data sets, using the CICS trace utility program, DFHTU410.
- The GTF data sets, using a CICS-supplied routine with the MVS interactive problem control system (IPCS).

The CICS trace utility program, DFHTU410

The CICS utility program, DFHTU410, extracts all or selected trace entries from the A or B auxiliary trace data set, and formats and prints the data. You specify the type of entries to be processed by this program on trace selection parameters supplied in either of the following:

- A PARM parameter on the EXEC PGM=DFHTU410 statement
- The DFHAXPRM data set.

You can specify that all entries are to be processed, or select entries for processing, for example entries:

- Written to the auxiliary trace data set within a specified period of time
- Written for a specified terminal
- With a specified trace identifier
- With specified trace entry sequence numbers⁶
- Associated with a specified transaction identifier
- Associated with a specific instance of a transaction identifier (task)
- Associated with a selected kernel task
- That are for exception trace only.

You can select which trace entries you want to highlight in your formatted output by specifying:

- The time interval between one trace entry and the next being written.

⁶ The sequence number is given in each trace entry, and can be determined from a summary trace point.

If more than the specified interval elapses before the next trace entry is written, then this next trace entry is formatted and printed with an asterisk (*) to draw your attention to this entry.

You can use the job control statements shown in Figure 34 to invoke the utility program for each auxiliary trace data set. Only use the trace utility program to print auxiliary trace data sets that you have opened in the most recent run of CICS. If you did not open an auxiliary trace data set during the most recent run of CICS, the trace utility program either prints records from a previous run or cannot recognize the records. If you opened the auxiliary trace data set A in the most recent run of CICS, but did not open auxiliary data set B, you can print data set A, but not print B.

```
//PRTRACE JOB accounting info,name,MSGLEVEL=1,CLASS=A,MSGCLASS=A, 1
//          REGION=2M
//PRINT EXEC PGM=DFHTU410
//STEPLIB DD DSN=CICS410.SDFHLOAD,DISP=SHR
//DFHAUXT DD DSN=CICS410.DFHxxxx,DISP=SHR 2 3
//DFHAXPRT DD SYSOUT=A
//DFHAXPRM DD *
:
[trace selection parameters] 4
/*
```

Figure 34. Sample JCL to print CICS trace data from an auxiliary trace data set

Notes:

1 The sample JCL gives a region size of 2MB that you might typically need to run the DFHTU410 utility. You can use the sample region size as a basis for your own JCL, but must ensure that the region size is large enough to run the DFHTU410 utility in your CICS environment.

2 Modify the DSN parameter to specify either the DFHAUXT or DFHBUXT data set, depending on whether the data is on the A or B data set. The ddname must be DFHAUXT for both the A and the B data set.

3 If your trace data sets are on tape, and the data set occupies more than one volume, you must begin with the first volume. The DD statement for trace data sets on tape might be as follows:

```
//DFHAUXT DD DSN=CICS410.DFHAUXT.,DISP=(OLD,KEEP),
//          VOL=SER=volid,UNIT=TAPE
```

4 You can define the number of lines to be printed and define which trace records that you want to print by specifying trace control statements, as described in “The trace selection parameters.”

The trace selection parameters

You code the trace selection parameters to define the number of lines to be printed on a page and to define which trace records you want to select for printing in the DFHAXPRM DD statement, or in the PARM parameter. For example:

```
//PRINT EXEC PGM=DFHTU410,PARM='selection_parameter,selection_parameter,...'
```

PAGESIZE=(value)

This statement specifies the number of lines printed on a page. You can specify a value in the range 20 through 9999 lines per page. If you specify an

incorrect value, CICS issues an error message and stops the trace. The default value is 55.

Note: This parameter is not valid for printing GTF trace entries.

ABBREV|FULL

Code this statement to indicate how much of each trace entry you want printed. If you specify this statement it must always be the first statement in either the PARM parameter or the DFHAXPRM data set.

ABBREV indicates that you require the abbreviated, one-line-per-entry, form of trace print.

FULL indicates that you want a fully formatted print of all the data in each entry. This is the default.

ALL

This statement specifies that all trace entries in the auxiliary trace data set are to be printed. This is the default.

ENTRY_NUM={({nnnnnn|nnnnnn-nnnnnn}[, {nnnnnn|nnnnnn},...,])

This statement specifies the sequence numbers of one or more trace entries that you want to print. Each sequence number can be up to six digits in length. If you specify a range of sequence numbers, by using xxxxxx-yyyyyy, the second sequence number (yyyyyy) must be larger than the first (xxxxxx).

EXCEPTION

This statement specifies that only exception trace entries in the auxiliary trace data set are to be printed.

Note: This parameter is not valid for printing GTF trace entries.

INTERVAL={00.128|number of seconds}

This parameter specifies the interval between auxiliary trace entries after which entries highlighted with an asterisk as follows:

- In abbreviated trace format, the asterisk appears to the left of the sequence number.
- In full trace format, the asterisk appears (as it does in releases prior to CICS/ESA 4.1 where a system-imposed time interval of 0.0128 seconds applies) as the next character after the printed time interval.

If successive auxiliary trace entries are written at intervals equal or greater than this limit, they are highlighted in the same manner.

If successive auxiliary trace entries are written at intervals less than this limit, they are not highlighted. They are, however, written, formatted and printed.

If you specify no INTERVAL value, a default of 0.128 seconds applies.

You can specify interval values in the range zero seconds (where all trace entries would be highlighted) through 99.999999999 seconds.

Note: The interval extends to ten decimal places. Zeros are padded from the right.

KE_NUM=(xxxx[,xxxx,,-])

This statement specifies that only the entries for tasks with the specified hexadecimal kernel task numbers are printed.

PAGESIZE=(value)

This statement specifies the number of lines printed on a page. You can specify a value in the range 20 through 9999 lines per page. If you specify an incorrect value, CICS issues an error message and stops the trace. The default value is 55.

Note: This parameter is not valid for printing GTF trace entries.

TASKID=({id|id-id}[,,{id|id-id},,...])

This statement specifies the task identifiers (id) of one or more tasks for which trace entries are to be printed. An id value can be in any of the following forms, to compare with the task field in the formatted trace data:

- Any number up to five decimal digits long
- Any of the character strings JAS, J01 through J99, III, TCP, or DSTCB
- Any non-numeric two-character domain ID of the attaching domain (for non-TCA) tasks.

You can specify a range of task identifiers of the five decimal digit form by using a hyphen (for example, TASKID=nnnnn-nnnnn).

TERMID=(tttt[,tttt],,...,)]

This statement specifies the terminal identifiers (tttt) of one or more terminals for which trace entries are to be printed.

If you use the TERMID parameter to specify the trace entries you want formatted, the DFHTU410 program selects all the trace entries that are associated with any transaction-attach trace entries it finds containing the terminal identifier(s) you specify. For more information about how trace entries for tasks are associated with transaction-attach trace entries, see “Identifying trace entries from their transaction-attach entries” on page 167.

TRANID=(tttt[,tttt],,...,)]

This statement specifies the transaction identifiers of one or more transactions for which trace entries are to be printed.

If you use the TRANID parameter to specify the trace entries you want formatted, the DFHTU410 program selects all the trace entries that are associated with any transaction-attach trace entries it finds that contain the transaction identifier(s) you specify. For more information about how trace entries for tasks are associated with transaction-attach trace entries, see “Identifying trace entries from their transaction-attach entries” on page 167.

TIMERG=(hhmmss-hhmmss[,hhmmss-hhmmss],,...,)]

This statement specifies the time period or periods for which trace entries are to be printed. Time periods are shown by pairs of values represented as hours (hh), minutes (mm), and seconds (ss) separated by a hyphen. The ending value of each pair must be later than the starting value.

The DFHTU410 program converts the store-clock (STCK) values in the trace entries to whole seconds prior to comparing against the time range you specify. Fractions of a second are ignored; that is, all times are rounded down to the nearest whole second, which means in effect that the minimum time span can be up to two seconds. For example, if you specify TIMERG=153000-153001, the DFHTU410 program includes all trace entries with times of 153000.00000000 to 153001.99999999 inclusive.

Note: This parameter is not valid for printing GTF trace entries.

TYPETR=({ddxxxx[ddxxxx-xxxx]}, {ddxxxx[ddxxxx-xxxx]})

This statement specifies the trace entry identifiers for the particular domain entries, specified by the domain id and a point id within the domain.

dd represents the domain identifier (listed in Table 14).

xxxx represents the point ID within the domain in the form of a four-character hexadecimal value (0000-FFFF). You can specify a range of point IDs by using a hyphen.

Table 14. CICS domains and their identifiers

Domain ID	Domain
AP	Application
DD	Directory manager
DM	Domain manager
DS	Dispatcher
DU	Dump
EX	External CICS interface
FT	Feature table
GC	Global catalog
KE	Kernel
LC	Local catalog
LD	Loader
LM	Lock manager
ME	Message
MN	Monitoring
PA	Parameter manager
PG	Program manager
SM	Storage manager
ST	Statistics
TI	Timer
TR	Trace
US	User
XM	Transaction manager
XS	Security manager

UPPERCASE

This statement specifies that you want the trace output in uppercase only. If you want trace output in mixed case (the default), do not code this parameter.

Identifying trace entries from their transaction-attach entries

The AP domain writes a trace entry each time a transaction is attached for execution. It is this transaction-attach trace entry that contains the terminal and transaction identifiers. It also contains the task identifier that is unique to a particular instance of a transaction. This is illustrated in the diagram shown in Figure 35 on page 169.

If you select trace entries by specifying the TRANID or TERMID parameters, the DFHTU410 program searches for any transaction-attach trace entries that contain the specified TERMID or TRANID. It then formats any associated trace entries, identified by the TASKID found in the transaction-attach trace entry data.

For example, if the entries in your auxiliary trace data set are as illustrated in Figure 35 on page 169, you can obtain formatted trace output for task IDs 00123 and 00124 by specifying either the TERMID or TRANID parameters. This is possible because the associated transaction-attach trace entries are present (see record numbers 2 and 7 in the diagram). However, you cannot obtain formatted trace output for task ID 00120 by specifying a TERMID or TRANID, because the auxiliary trace data does not contain the transaction-attach trace entry for that task.

Rules for coding trace selection parameters

If you enter the control statements in the DFHAXPRM data set, enter them in columns 1 to 71 of the 80-character records. Leading blanks are ignored, and no imbedded blanks are allowed. The first blank in a line terminates the statements on that line; you can include comments after the first blank. The TERMID and TRANID entries are padded with blanks to four characters if necessary.

You can specify each control statement one or more times; for example,
TASKID(xxxx,zzzz,yyyy,aaaa,bbbb,cccc,dddd,eeee,ffff,gggg,hhhh,iiii,jjjj),
TASKID(kkkk,rrrr-uuuu,www)

You must use commas to separate keywords and entries in a list. Continuation to another record is allowed after any comma **that separates keywords**, provided the comma is in column 71 or is followed by a blank. Continuation records can start in any column.

For example, the following statements can be coded in DFHAXPRM:

```
TERMID=LP1,           [Select entries for terminal LP1  
TRANID=(ABRW,AORD,MYTR), [Select entries for tranids ABRW, AORD, & MYTR  
TIMERG=(123000-150000) [Select entries timed between 1230 and 1500
```

The same example could be coded in the PARM parameter as follows:

```
//      EXEC PGM=DFHTU410,PARM=('TERMID=LP1,TRANID=(ABRW,AORD,MYTR) ',  
//      'TIMERG=(123000-150000)')
```

Note: The following example would not work, because the list has been split *within* the keyword rather than between keywords:

```
TRANID=(ABRW,AORD, [Select entries for tranids ABRW, AORD,  
MYTR),              & MYTR
```

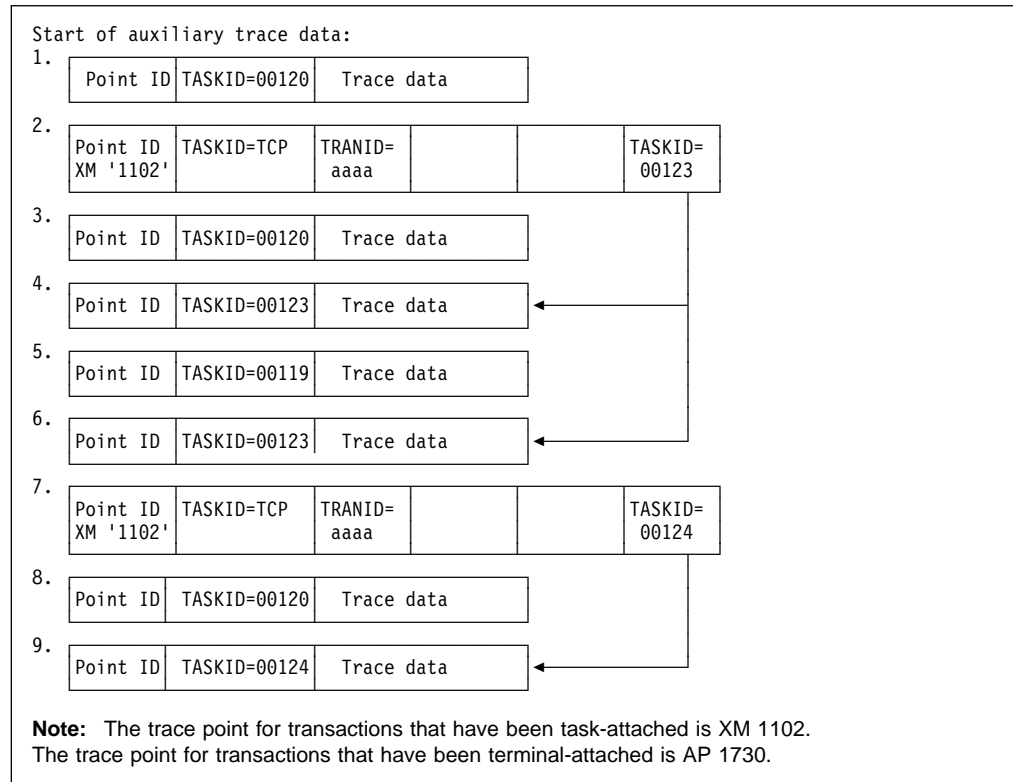


Figure 35. Association of transaction-attach trace entries with task entries

Using IPCS to print trace records written to GTF

When GTF trace is on, and the CICS master trace flag is also on, CICS writes trace entries to a GTF data set. CICS writes these records by issuing the GTRACE macro with the following parameters specified:

FID=X'EF' The format identifier (FID) of the CICS GTF trace entry

ID=X'F6C' The subsystem event trace identifier for CICS GTF trace entries.

You can print CICS trace entries written to GTF by invoking IPCS with the GTFTRACE subcommand, and specifying the USR parameter with the event trace identifier of the records you want IPCS to select for formatting. You can also specify most of the DFHTU410 selective trace control statements on the CICS(text) parameter.

PN92508

CICS410.SDFHLOAD changed to CICS410.SDFHLINK by APAR PN92508.

The CICS-supplied formatting routines are called DFHTG410 and DFHTR410, supplied in CICS410.SDFHLINK. DFHTG410 has the alias of AMDUSREF. The last two characters of the AMDUSREF alias ("EF") correspond to the format identifier (FID), and enable IPCS to invoke the CICS formatting routine automatically when you use the GTFTRACE subcommand.

Several CICS regions, at different CICS releases, can write to the same GTF data set. You can print CICS/ESA 4.1, CICS/ESA 3.3, and CICS/ESA 3.2.1 GTF trace entries written to GTF using the same GTFTRACE command. To do so, you must make available the formatting routine for CICS/ESA 4.1 (DFHTG410, its alias AMDUSREF, and DFHTR410) and the formatting routine for CICS/ESA 3.3 and 3.2.1, called DFHTRVR3.

PN92508

CICS330.SDFHLOAD changed to CICS330.SDFHLINK by APAR PN92508.

The DFHTRVR3 module is provided by APAR PN59159 for CICS330.SDFHLINK or CICS321.SDFHLOAD. The version of DFHTRVR3 in CICS330.SDFHLINK can be used to format GTF trace entries for CICS/ESA 3.3 and CICS/ESA 3.2.1, but may not format all CICS/ESA 3.2.1 trace entries correctly. The version of DFHTRVR3 in CICS321.SDFHLOAD formats only GTF trace entries for CICS/ESA 3.2.1.

The GTFTRACE subcommand of IPCS and associated parameters

To use IPCS to format and print CICS trace entries, you must specify the GTFTRACE subcommand. The following are the main IPCS GTFTRACE parameters that you need to process CICS trace entries:

GTFTRACE

Use this IPCS subcommand to format CICS trace records contained in a GTF trace data set.

JOBNAME({name1[,name2,..,namen]})

Code this to specify one or more jobnames for which you want DFHTR410 to format trace entries.

CICS((CICS trace selection parameters))

Code this parameter to specify any selection and formatting control statements required by the CICS formatting routine, DFHTR410. You can code any of the DFHTU410 parameters except EXCEPTION, PAGESIZE and TIMERG, which are not allowed. You can change the default pagesize by using the IPCS subcommand PROFILE PAGESIZE(value); for an example, see Figure 36 on page 171. For further information about the PROFILE PAGESIZE subcommand, see the *MVS/ESA IPCS Command Reference* manual. (You can use the GTFTRACE START and STOP parameters in place of TIMERG; see below.)

Note: The whole string of CICS trace selection parameters must be enclosed in parentheses. If your CICS trace selection parameter is more than can be contained on one line, terminate the line with a right parenthesis followed by a comma, and specify the remainder on the next line. You must repeat the CICS keyword on the continuation line(s).

START(ddd, hh.mm.ss) and STOP(ddd, hh.mm.ss)

Code the START and STOP parameters to specify trace entries for a particular time range. If you omit the STOP parameter, IPCS continues processing until it reaches the end of the data set.

USR(event-id-value-list|ALL)

Code this parameter to specify formatting of subsystem event trace records created by the GTRACE macro. The trace ID for CICS GTF trace entries is

'CICS', which translates to X'F6C'. For information about the IDs of other subsystem trace records (for example, VSAM, VTAM), see the *MVS/ESA IPCS Command Reference* manual. (You can code X'F6C' directly for the CICS trace event ID; USR(CICS) is an alias for USR(F6C).)

Specify ALL to request formatting of all subsystem trace entries.

There are many other parameters that you can specify on the GTFTRACE subcommand of IPCS. For information about the GTFTRACE command, see the *MVS/ESA IPCS Command Reference* manual.

Sample batch job to print CICS GTF trace entries

You can specify the GTFTRACE subcommand of IPCS in TSO, or in a batch job as shown in Figure 36.

```

//IPCSGTF JOB ( accounting info),
//          CLASS=A,MSGCLASS=A,MSGLEVEL=(1,1),NOTIFY=userid
//PRINTTR EXEC PGM=IKJEFT01,REGION=4096K
//STEPLIB DD DSN=CICS410.SDFHLINK,DISP=SHR           1
//          DD DSN=CICS330.SDFHLINK,DISP=SHR
//SYSTSPRT DD DUMMY
//DFHTRACE DD DSN=GTF.TRACEnn,DISP=SHR
//IPCSDDIR DD DSN=ipcs.dump.directory,DISP=SHR      2
//SYSABEND DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//SYSTSPRT DD SYSOUT=A
//IPCSPRNT DD SYSOUT=A
//SYSTSIN DD *
IPCS NOPARM
SETDEF DD(DFHTRACE) NOPROBLEM PRINT NOCONFIRM NOTERMINAL LIST
PROFILE PAGESIZE(60)
GTFTRACE +
CICS(( ABBREV,TERMID=LP1,TRANID=<ABRW,AORD>,)) +
CICS(( TYPETR=(SM01FF-03BC),TASKID=(J01,J03-J05,TCP))) +
USR(F6C)
END
/*
//
  
```

Figure 36. Sample IPCS job to print CICS trace entries from a GTF data set

Notes for: Figure 36

1 The STEPLIB statement includes the load libraries that contain the modules (DFHTG410, its alias AMDUSREF, DFHTR410 and, if needed, DFHTRVR3) to be used to format the GTF trace entries. Depending on which releases of CICS have GTF trace entries to be printed, you should include the following libraries in the STEPLIB concatenation unless you have included them in the MVS linklist:

PN92508

SDFHLOAD changed to SDFHLINK in the following table by APAR PN92508.

STEPLIB	CICS releases to be formatted
CICS410.SDFHLINK	CICS/ESA 4.1 only
CICS410.SDFHLINK CICS330.SDFHLINK	CICS/ESA 4.1, 3.3, and 3.2.1; although not all CICS/ESA 3.2.1 trace entries may format correctly.
CICS410.SDFHLINK CICS321.SDFHLOAD	CICS/ESA 4.1 and 3.2.1

+ **2** The IPCSDDIR statement specifies a directory data set needed by IPCS. A
+ sample job to create an IPCS directory data set is given in Figure 43 on page 194.

Invoking the CICS formatting routine from TSO

PN92508

The following paragraph has been changed by APAR PN92508.

| To enable the CICS formatting routines to be invoked from TSO, copy the modules
| DFHTG410, AMDUSREF, DFHTR410, and DFHTRVR3 to a suitable library in the
+ MVS linklist, or ensure that the CICS libraries listed above are included in the MVS
+ linklist.

| Ensure that you have sufficient TSO storage (specified for your TSO userid when
| you logon) for GTF trace formatting, because otherwise storage fragmentation
| causes formatting problems. When using one job to format trace entries for several
| CICS releases, the loading of the multiple release formatters needed uses more
| storage.

Chapter 12. Dump utility programs

CICS produces two types of dump: transaction dumps that CICS writes to CICS transaction dump data sets; and MVS system dumps (SVC dumps) via the MVS SDUMP macro, and written to SYS1.DUMPnn data sets. CICS provides two utility programs to help you analyze these dumps. These are:

1. The transaction dump utility program, DFHDU410, for processing CICS transaction dumps. For details of this CICS batch utility program, see “The CICS transaction dump utility program, DFHDU410.”
2. The dump exit that is invoked via the interactive problem control system (IPCS) for processing either MVS SDUMP dumps that CICS takes, or CICS jobs in SVC dumps that are taken by the MVS dump command.

Note: For the CICS-supplied IPCS dump exit routine to format an SDUMP successfully, certain SDUMP options must be in force at the time the dump is taken. (See page 180.)

You can use IPCS either interactively or from an MVS batch job. For more information about using IPCS, see “Using IPCS to format and analyze CICS dumps” on page 179.

The CICS transaction dump utility program, DFHDU410

The output from the CICS dump domain is written to DASD or tape, depending on which you specified when the transaction dump data sets were created and defined to CICS. The transaction dump utility program, DFHDU410, prepares the transaction dump output for printing and prints the formatted information.

Selecting transaction dump output for the DFHDU410 program

You govern the selection of the dumps that you want printed by control statements in the SYSIN data set. In addition to the dumps you select, the DFHDU410 program always writes a summary in the form of an index of the dumps that are on the data set being processed. The index information is taken from transaction dump header records, and is written either to a data set defined with a DD name of DFHTINDX, or to the SYSPRINT data set if the DFHTINDX DD statement is missing.

Format of the SYSIN control statements

```
SELECT TYPE={OR|NOTOR|AND|NOTAND|SCAN}  
[TRANID=({value|generic-value}[,{value|generic-value}],...)]  
[DUMPCODE=({value|generic-value}[,{value|generic-value}],...)]  
[DUMPID=({value|value-range}[,{value|value-range}],...)]  
[PAGESIZE=(value)]  
[TIME=({time|time-range}[,{time|time-range}],...)]  
[UPPERCASE=YES]  
END
```

Figure 37. SYSIN control statements for the DFH DU410 program

If you do not define a SYSIN data set, or SYSIN does not contain any control statements, all dumps in the DFHDMPDS data set are printed.

Code only one SELECT statement and one END statement, but you may code multiple TRANID, DUMPCODE, DUMPID, and TIME control statements. Each control statement must be on a separate line, but can start in any column. For example:

```
//SYSIN DD *  
SELECT TYPE=OR  
DUMPCODE=(MY*)  
TRANID=ABCD  
END
```

The descriptions of the statements you can code in SYSIN are as follows:

SELECT TYPE={OR|NOTOR|AND|NOTAND|SCAN}

This control statement, which is mandatory if you are specifying any of the other selection control statements, must be the first in SYSIN. Code the TYPE parameter with one of the following selection operands:

OR Print only those dumps that match at least one of the fields defined in any TRANID, DUMPID, DUMPCODE, or TIME control statements that follow the SELECT statement. This is the default if you omit the TYPE parameter.

NOTOR Print only those dumps that do *not* match any of the fields defined in any TRANID, DUMPID, DUMPCODE, or TIME control statements that follow the SELECT statement.

AND Print only those dumps that match all of the fields defined in any TRANID, DUMPID, DUMPCODE, or TIME control statements that follow the SELECT statement.

NOTAND Print only those dumps that do *not* match the combination of the fields defined in any TRANID, DUMPID, DUMPCODE, or TIME control statements that follow the SELECT statement.

SCAN Do not print any dumps, but write only the summary, either to the DFHTINDX data set, or to the SYS PRINT data set if the DFHTINDX DD statement is missing. If you code SCAN, all other statements in the SYSIN dataset (apart from the END statement) are ignored.

If you code any of the following control statements, they must appear in the SYSIN data set after a SELECT statement, and before the END statement. Each control statement must be on a separate line, but can start in any column.

TRANID=({value|generic-value}[,value|generic-value],,..)

Code this control statement to select dumps by their transaction identifier (ID), coding one or more four-character transaction IDs. You can code up to 20 transaction IDs on the TRANID statement(s); excess transaction IDs are ignored. Code the transaction IDs either as explicit IDs, or as a generic form using plus (+) or asterisk (*) symbols as arbitrary characters. If you code a transaction ID of less than four characters, and without any arbitrary characters, it is assumed to be filled with trailing blanks (up to the limit of four characters for a transaction ID).

A + symbol represents any single character other than blank, and should be used to specify a single arbitrary character. For example:

TRANID=ABC

specifies a three-character transaction ID of 'ABC'.

TRANID=AB+

specifies a three-character transaction ID, where the first two are 'AB', and the third is any character (other than blank).

TRANID=CD+F

specifies a four-character transaction ID, where the first two are 'CD', the third is any character (other than blank), and the fourth is 'F'.

An asterisk (*) symbol represents any character string not containing blanks, for example:

TRANID=XY*

specifies a transaction ID, where the first two characters are 'XY', the third character can be any character other than a blank, and the fourth can be any character.

All of the above examples can be coded on the following TRANID statement:

```
TRANID=(ABC,CD+F,XY*,AB+)
```

DUMPCODE=({value|generic-value}[,{value|generic-value}],,..)

Code this to select dumps by a transaction dump code, which is either the four-character abend code or your own explicitly defined code if you requested the dump. You can code up to 20 dump codes on the DUMPCODE statement(s); excess dump codes are ignored. Code the dump codes either as explicit codes, or as a generic form using plus (+) or asterisk (*) symbols as arbitrary characters. See the TRANID control statement for details of how to use the arbitrary character symbols.

DUMPID=({value|value-range}[,{value|value-range}],,..)

Code this to select dumps by a 6- to 9-character dump identifier. You can code up to 10 dump identifiers or ranges of dump identifiers on the DUMPID statement(s); excess dump identifiers are ignored. The format of a dump identifier is xxxx/yyyy where xxxx represents the dump run number, and yyyy is the dump count. You must code the slash (/) symbol as a separator character between the dump run number and the dump count.

Note: The DFHDU410 program checks only that the DUMPID operand is valid in length, and contains only numeric and / characters. If you specify a wrong numeric dump run number or dump count, or specify the wrong number of / characters, the DFHDU410 program fails to find a matching dump.

The dump identifier operands are defined as follows:

Dump run number

A number in the range 1 to 9999. (Leading zeros are not used for this number, which is why the dump id can vary from 6 to 9 characters.) The dump run number begins at 1 when you first start CICS with a newly-initialized local catalog, and is incremented by 1 each time you restart CICS.

Note: The dump run number is saved in the local catalog when you perform a normal shutdown, but is reset if you start CICS with a START=COLD system initialization parameter.

Dump count

A number in the range 0001 through 9999. (Leading zeros are required in the dump id.) This is the number assigned to the dump in this run of CICS, starting at 0001 for the first dump, and incremented by 1 with each dump taken.

You can code the DUMPID parameter as a single value, as a range of values, or as a combination of both. If you specify a range of DUMPIDs, you must specify the lower value first. For example:

DUMPID=10/0005

This specifies a single dump identified as the fifth dump taken during dump run number 10.

DUMPID=125/0001-125/9999

This specifies all the dumps taken during dump run number 125.

DUMPID=(125/0001-125/0003,125/0019)

This specifies the first three dumps taken during dump run number 125, plus dump count number 19.

PAGESIZE=(value)

Code this parameter with the number of lines printed on a page. You can code values in the range 20 through 9999 lines per page. If you specify an incorrect value, CICS issues an error message and uses the default page size. The default value is 60.

TIME=({time|time-range}[,{time|time-range}],...)

Code this parameter to select dumps by the time at which a dump was taken. You can code up to ten time values or range of times on the TIME statement(s); excess times are ignored. Code either a time value or a range of times, or any combination of both, specifying the time in hours and minutes only, ignoring the seconds. (If CICS takes more than one transaction dump in the same minute, all dumps matching the hour and minute are selected.)

The format for time is hh.mm or hh:mm, and you specify a range of times as hh.mm-hh.mm or hh:mm-hh:mm. You must specify the hours and minutes as two digits, in the range 00 through 24 and 00 through 59 respectively.

UPPERCASE=YES

Code this parameter if you want the data output in uppercase only. The parameter must be coded as shown in uppercase characters with no spaces between words. If you want output in mixed case (the default), do not code this parameter.

END

This statement is optional and terminates the SELECT group. All statements following the END statement are ignored. If you omit the END statement, the SELECT group is terminated by the end of the SYSIN data set.

Job control statements to run the DFHDU410 program

The job stream to run the DFHDU410 program should include DD statements for the following data sets:

ddname	Description
DFHDMPDS	(Mandatory) The input data set, from which the dump data is to be processed.
DFHPRINT	(Mandatory) The output data set, usually a printer, to which the dump data is written.
DFHTINDX	(Optional) The output data set to which the dump index summary is written, needed if you want the index summary output. If you omit the DFHTINDX DD statement, DFHDU410 tries to open a SYSPRINT data set for dump index output. If unsuccessful, you get message IEC130I, and the dump index summary is not written.
SYSIN	(Optional) The SYSIN data set in which you define the control statements for the DFHDU410 program, needed if you want to code specific selection parameters. If you omit the SYSIN DD statement, you get message IEC130I.

See Figure 38 for a sample job stream for the DFHDU410 program.

```
//PRNTDMP JOB accounting info,name,MSGLEVEL=(1,1),
//          CLASS=A,MSGCLASS=A,REGION=2M
//          EXEC PGM=DFHDU410,PARM='command,command,...'
//STEPLIB DD DSN=CICS410.SDFHLOAD,DISP=SHR
//DFHDMPDS DD DSN=CICS410.DFHMPA,DISP=SHR
//DFHTINDX DD SYSOUT=A
//DFHPRINT DD SYSOUT=A,DCB=(BLKSIZE=133)
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
SELECT TYPE=OR
[selection parameters, each on a separate line]
END
/*
//
```

Figure 38. Sample job to format and print CICS transaction dump data sets

1 The sample JCL gives a region size of 2MB that you might typically need to run the DFHDU410 utility. You can use the sample region size as a basis for your own JCL, but must ensure that the region size is large enough to run the DFHDU410 utility in your CICS environment.

To run the transaction dump utility program concurrently with CICS to process the inactive disk transaction dump data set, specify DISP=SHR in the DD statements defining the transaction dump data sets in the startup job stream.

The PARM options of the EXEC statement are:

+ **NOABBREV**
+ Prevents the trace entries from being formatted in the abbreviated format in the
+ transaction dump.

+ **NOFULL**
+ Prevents the trace entries from being formatted in the full format in the
+ transaction dump.

+ Trace formatting may be suppressed by using both the NOABBREV and
+ NOFULL options. The default action is that the trace will be printed in
+ ABBREVIATED followed by FULL formats (see the ABBREV and FULL
+ keywords of the DFHTUP utility program for details).

+ **Note:** ABBREV and FULL are not valid keywords of the DFHDUP utility
+ program.

DOUBLE|SINGLE

For SINGLE, the transaction dump output is printed single-spaced. For
DOUBLE the output is printed with a blank line between the printed lines.

TRANSLATE=LC|FOLD|UC

For LC, lowercase letters are printed as lowercase and uppercase letters as
uppercase in the interpreted output on the right side of the dump output. For
FOLD, all lowercase letters are converted to uppercase in the interpreted
output. For UC, only the uppercase letters are printed in the interpreted output.

+ The contents of a transaction dump data set are never erased, but they may be
+ overwritten when the data set is next opened for use. This happens only when:

- The data set is opened during initialization.
- You switch to the data set by using the CEMT SET DUMPDS SWITCH command, or by the corresponding EXEC CICS SET command.
- The data set is opened explicitly by the CEMT SET DUMP OPEN command, or by the corresponding EXEC CICS SET command.

+ Any dumps taken during the current run of CICS overwrite the contents of the data
+ set in use.

+ When a dump data set is closed, an EOF marker is added after the last dump
+ record written. If you switch from one dump data set to another without taking a
+ dump, the EOF marker is added as the first record of the closed dump data set.

+ If you use the dump utility program to print a dump data set that is not in use by
+ CICS (that is, has been opened during the current run but is now closed), any
+ dumps written during the current run are printed (up to the EOF marker).

+ If you use the dump utility program to print a dump data set that is still in use by
+ CICS (that is, before the data set has been closed), any transaction dumps written
+ during the current run are printed. These may be followed by an unidentified partial
+ transaction dump from a previous run, whose header has been overwritten during

- + the current run. Any such partial transaction dumps may be followed by further transaction dumps from the previous run (up to the EOF marker from that run).
- + Do not use the dump utility program to print a dump data set that has not been
- + opened during the current run of CICS. If you try to, either transaction dumps from
- + a previous run are printed, or the program is unable to recognize the records on the data set.

Using IPCS to format and analyze CICS dumps

The interactive problem control system (IPCS) provides MVS installations with an interactive facility for diagnosing software failures. MVS SDUMPs can be produced by CICS or by entering the MVS DUMP command. You can use IPCS to format and analyze these SDUMPs or to analyze stand-alone dumps obtained when CICS was active in the system being dumped. You can view the dumps at your terminal or you can print them.

The IPCS dump analysis subcommands enable you to:

- Examine the data in a dump
- Locate and verify control blocks associated with certain functions or system components
- Trace and verify chains of control blocks
- Perform contention analysis on key MVS resources
- Locate modules and unit control blocks (UCBs)
- Execute user-written exits for certain control blocks
- Keep a list of the names and locations of control blocks and areas of dump that you consider important.

To enable you to analyze CICS SDUMPs written to SYS1.DUMPnn data sets by the SDUMP macro, you can use the IPCS VERBEXIT subcommand to execute a CICS-supplied IPCS dump exit. This dump exit enables you to:

- Process a dump selectively by specifying one or more CICS component identifiers as parameters to the exit.
- Select parts of the CICS internal trace table to format for a system dump. How you do this is described in “Selecting parts of the CICS internal trace table” on page 182.

For further information about IPCS, see the *MVS/ESA IPCS User's Guide*.

What you must do before you can use IPCS to format CICS SDUMPs

Before you can use IPCS to format CICS SDUMPs, you must:

- Ensure that certain SDUMP options are in force when the dump is taken. (See “The SDUMP options needed to support the CICS dump exit” on page 180.)
- Ensure that the DFHIPCSP member can be found by your IPCS job. (See “Specifying DFHIPCSP CICS exit control data needed” on page 180.)
- Rename any dump exit routines from earlier CICS releases to ensure that the exit control data in the DFHIPCSP member corresponds to the dump exit

routine names. (See “Renaming dump exit routines for releases before CICS/ESA 4.1” on page 180.)

- Ensure that the CICS-supplied dump exit routines can be found by your IPCS job. (See “Making available CICS dump exit routines needed” on page 181.)

The SDUMP options needed to support the CICS dump exit

The CICS dump exit is unable to format a CICS dump successfully unless the minimum SDUMP options are in force at the time the dump is written. CICS issues an SDUMP macro request specifying a number of SDUMP options, of which the following must not be overridden:

ALLPSA	Prefix storage area for all processors
CSA	Common storage area.
GRSQ	Global resource serialization queues
LPA	Link pack area
NUC	Non-page-protected areas of the DAT-on nucleus
RGN	Entire private area of the CICS address space
SQA	System queue area
SUMDUMP	The summary dump function
TRT	GTF, system trace, and master trace data

If you set the dump mode for SDUMP to override mode (using the MVS CHNGDUMP SET OVER command), you must ensure that at least these options are set in the system’s SDUMP options list.

Specifying DFHIPCSP CICS exit control data needed

IPCS provides an exit control table with imbed statements to enable other products to supply exit control information. The IPCS default table, BLSCECT, normally in the SYS1.PARMLIB library, has the following entry for CICS:

```
# IMBED MEMBER(DFHIPCSP) ENVIRONMENT(ALL) /* CICS */
```

The CICS-supplied DFHIPCSP member, installed in the CICS410.SDFHPARM library, contains the CICS release-specific entries for the IPCS exit control table. These entries are listed in Figure 39 on page 181.

You must ensure that this DFHIPCSP member can be found by your IPCS job. You can either copy the DFHIPCSP member into the SYS1.PARMLIB library (so that it is in the same default library as BLSCECT) or provide an IPCSPARM DD statement to specify the library containing the IPCS control tables. For example:

```
//IPCSPARM DD DSN=SYS1.PARMLIB,DISP=SHR For BLSCECT
// DD DSN=CICS410.SDFHPARM,DISP=SHR For DFHIPCSP
```

The names of the IPCS exit routines specified by the EP(name) operands in the DFHIPCSP member must match the names of the CICS-supplied release-specific IPCS exit routines.

Renaming dump exit routines for releases before CICS/ESA 4.1

In CICS/ESA 4.1, the name of the IPCS dump exit routine matches the corresponding exit control data in the DFHIPCSP member.

In earlier CICS releases, the CICS formatting routine for use under the IPCS is supplied as DFHPDX. This standard name is not suitable for those users running more than one release of CICS, because the dump formatting process in each

version of DFHPDX is release-specific, and you must use the correct version for the system dump you are formatting. To use the DFHIPCSP member as supplied, rename the versions of the DFHPDX routines supplied with earlier CICS releases to the names shown.

```

/* ===== */
EXIT EP(DFHPD212) VERB(CICS212) ABSTRACT(+
'CICS Version 2 Release 1.2 analysis')
EXIT EP(DFHPD321) VERB(CICS321) ABSTRACT(+
'CICS Version 3 Release 2.1 analysis')
EXIT EP(DFHPD330) VERB(CICS330) ABSTRACT(+
'CICS Version 3 Release 3 analysis')
EXIT EP(DFHPD410) VERB(CICS410) ABSTRACT(+
'CICS Version 4 Release 1 analysis')
/* ===== */

```

Figure 39. Release-specific entries in DFHIPCSP for DFHPDnnn routines

Modifying IPCS jobs on releases before CICS/ESA 4.1

IPCS jobs on CICS releases before CICS/ESA 4.1 used the VERBEXIT parameter CICSDATA. To enable your IPCS jobs on those earlier releases to use the renamed dump exit routines, you must change occurrences of the VERBEXIT parameter CICSDATA in those jobs to refer to the new verb names for the CICS release that produced the dump data. (See Figure 39 for the CICS-supplied verb names.) For example, in a CICS/MVS 2.1.2 IPCS job change “VERBEXIT CICSDATA ...” to “VERBEXIT CICS212”

Making available CICS dump exit routines needed

The dump exit routine that you use to format CICS system dump data must be the routine that was supplied with the release of CICS that produced dump data.

To ensure that your IPCS job can find the appropriate dump exit routine to format the CICS system dump data, you should add the library containing the dump exit routine to the MVS linklist. The dump exit routine for CICS/ESA 4.1, DFHPD410, is installed in the SYS1.CICS410.SDFHLINK library along with other modules needed in the MVS linklist. This routine is named with the release identifier as part of the name; that is, DFHPD**410**.

To enable the CICS-supplied IPCS dump exit routines, all called DFHPDX, from earlier CICS releases to be used, you should:

- Rename the routines to use the release identifier as part of the name (See “Renaming dump exit routines for releases before CICS/ESA 4.1” on page 180.)
- Copy the routines to the SYS1.CICS410.SDFHLINK library or another suitable library in the MVS linklist.

Selecting parts of the CICS internal trace table

You can select which parts of the CICS internal trace table to format for a system dump, by using a new CICS dump exit parameter, TRS, for the IPCS SDUMP formatting program: This parameter enables you to select trace entries by:

- Kernel task
- Task identifier
- Terminal
- Transaction identifier
- Time period
- Trace identifier.

To select the parts of the internal trace to be formatted by IPCS, you specify the TRS parameter on the IPCS VERBEXIT command, for example,

```
VERBEXIT CICS410 'DEF=1,DLI=1,KE=3,TR=2,TRS=<<TRANID=CSSC,KE_NUM=12>'
```

Notes:

1. The VERBEXIT statement specifies the verb name CICS410 to process CICS/ESA 4.1 system dump data. This corresponds to the IPCS dump exit routine DFHPD410, as specified in the DFHIPCSP member in the CICS410.SDFHPARM library.
2. For the TRS parameter to work, you must also specify the TR parameter, without a value of 0, to use output from the trace domain.

For more information about the statements that you can use to select parts of the CICS internal trace table, see “The trace selection parameters” on page 164.

Using CICS-supplied dump exit routines to format CICS SDUMPS

To use IPCS to format a CICS SDUMP, specify the CICS dump exit parameters on the VERBEXIT subcommand of IPCS, using the verb name for the CICS release that produced the dump data. (See Figure 39 on page 181 for the CICS-supplied verb names.)

The syntax of the CICS exit parameters is shown in Figure 40 and described in “The CICS dump exit parameters” on page 183.

For some examples of using IPCS to process CICS SDUMPs, see page 184.

```
[JOB={jobname|CURRENT}]
[UPPERCASE]

[,DEF={0|1|2|3}]
[,keyword [=levelnumber]]
  where – keyword specifies the CICS component ID (see page 185
         for a full list of all the keywords).
         – levelnumber specifies the level of data to be output,
           either to a terminal or to a printer (see page 185 for details of
           the available level numbers).
```

Figure 40. The CICS dump exit parameters

The CICS dump exit parameters

The CICS dump exit parameters are as follows:

JOB={jobname|CURRENT}

This optional parameter specifies which job in the dump is to be formatted (when there is more than one job in the dump). You should specify this parameter only if you know that the dump that you are processing contains more than one job. If the dump was taken by CICS via the SDUMP macro *for a dump code which is defined as RELATED*, the dump may contain more than one CICS job, so you should specify the JOB parameter. If the dump code is defined as LOCAL, the dump contains only one CICS job, so you can omit the JOB parameter and still process only that one dump.

jobname Formats the job identified by **jobname**

CURRENT Formats all the CICS jobs in the dump data set.

If you omit the JOB parameter, all the CICS jobs found in the dump data are formatted.

UPPERCASE

This optional parameter specifies that you want the dump data output in uppercase only. If you want output in mixed case (the default), do not code this parameter.

DEF={0|1|2|3}

This parameter specifies a default level for the formatting of data from the dump data set. The DEF parameter is effective only for those components that are not included in a list of dump component keywords.

The possible levels that you can specify are as follows:

Level Meaning

0 Suppress

For those components not in a specified list of keywords, suppress all component formatting. If you specify DEF=0, but do not specify any component keywords, you still get the dump summary and, if appropriate, the error message index.

1 Summary

For those components not in a specified list of keywords, and where applicable, produce only a formatted summary from the control blocks. (A summary is not available for all components; see the level numbers available for the individual keywords for which a summary of dump information is available.)

2 Full

For those components not in a specified list of keywords, format all the control block information in full.

3 Summary and full

For those components not in a specified list of keywords, format all control blocks and (where applicable) the summary information.

The effects of omitting the DEF parameter are as follows:

- If you omit the DEF parameter and **do not** specify any component keywords, the result is as if you specified DEF=3. For example:
 - VERBEXIT CICS410 'JOB=CURRENT' formats all the available summary and control block information for the currently dispatched job(s). For the trace (TR) component, both the abbreviated and full trace are produced. The control block index is produced (see the IND component keyword) and, if appropriate, an error message index.
- If you omit the DEF parameter **and** specify one or more component keywords, the result is as if you specified DEF=0. For example:
 - VERBEXIT CICS410 'KE=1' produces a summary of the kernel dump data, plus the dump summary and, if appropriate, and error message index, but suppresses formatting of data for all other dump components.

Exceptions to the scope of the DEF parameter

The two parts of a CICS system dump that are not governed by component keywords, and are therefore outside the scope of the DEF parameter, are:

1. The dump summary.
2. The error message index.

The dump summary is always formatted, even if you specify DEF=0 and no component keywords. The error message index is produced only if an error or information message is output while the CICS dump exit is formatting the dump data, even if you specify DEF=0 and no component keywords. For example:

VERBEXIT CICS410 'DEF=2,DS=0' suppresses formatting of the dispatcher (DS) domain; the dump summary is formatted, and all other components are formatted for level 2 only. The error message index is only produced if an error or information message is output while the CICS dump exit is formatting the dump data.

For details of the dump summary and the error message index, see page 192.

Examples of the use of level numbers

VERBEXIT CICS410 'JOB=CURRENT,KE=1,DS' prints a summary of the kernel domain data, and all available information for the dispatcher domain.

VERBEXIT CICS410 'JOB=CURRENT,DEF=2,KE=1,DS=0' prints a summary of the kernel domain data, and the control blocks for all other components except for the dispatcher domain, which is suppressed.

VERBEXIT CICS410 without any parameters produces summary and control block output for all the CICS components in the dump.

The CICS410 dump exit component keywords

The component keywords specify which functional areas of the CICS dump you want the CICS410 exit to format, and the level number operand specifies the amount of data you want formatted. If you omit all of the component keywords, and provided you have not specified DEF=0, the CICS dump exit formats dump data for all components.

The syntax of the component parameter is as follows:

component-keyword[=0**|**1**|**2**|**3**]**

Specify the component keyword from the list of available keywords shown on page 185. The level number operand has the same function as the level number on the DEF parameter, but with different default rules, as follows:

- 0** **Suppress** all output for the component.
- 1** **Summary** only, but available only for certain components. If you code level 1 for a component that does not have a summary, it defaults to level 0 (that is, all formatting is suppressed). Note that level 1 has a special meaning for the trace and index components, as follows:
 - TR** If you specify level 1 for the TR (trace) component, you get the abbreviated trace only (see the ABBREV keyword on the DFHTU410 utility program for details).
 - IND** If you specify level 1 for the IND (index) keyword, you get the control block index sorted by address.
- 2** **Full** control block formatting. Level 2 has a special meaning for the trace and index components, as follows:
 - TR** If you specify level 2 for the TR (trace) component, you get full trace output (see the FULL keyword on the DFHTU410 utility program for details).
 - IND** If you specify level 2 for the IND (index) keyword, you get the control block index sorted by name.
- 3** **Summary and full**, that is, both level 2 and (where available) level 1. If you code level 3 for a component that does not have a summary, it defaults to level 2.

Note: If you omit the level number, it defaults to level 3 for those components that have a summary, and level 2 for those that do not.

#

AI [={0|2}**]**

Autoinstall model manager.

- 0** Suppress formatting of AI control blocks.
- 2** Format AI control blocks.

AP [={0|1|2|3}**]**

Application domain.

- 0** Suppress formatting of AP control blocks.
- 1** Format a summary of addresses of the AP control blocks for each active transaction.
- 2** Format the contents of the AP control blocks for each active transaction.

```

#           3           Format level-1 and level-2 data.
#
# CC [= {0|2}]
#           CICS catalog domain.
#
#           0           Suppress formatting of CC control blocks.
#           2           Format the CC control blocks.
#
# CP [= {0|2}]
#           Common programming interface.
#
#           0           Suppress formatting of CP control blocks.
#           2           Format the CPI static storage.
#
# CSA [= {0|2}]
#           CICS common system area.
#
#           0           Suppress formatting of the CSA.
#           2           Format the CSA and its extension, the optional features list
#                       (CSAOPFL).
#
# DD [= {0|1|2|3}]
#           Directory manager domain.
#
#           0           Suppress formatting of DD control blocks.
#           1           Format the directory manager summary.
#           2           Format the directory manager control blocks, including the anchor
#                       block, directory headers, and AVL tree headers.
#           3           Format level-1 and level-2 data.
#
# DLI [= {0|2}]
#           CICS DL/I interface.
#
#           0           Suppress formatting of DLI control blocks.
#           2           Format DLI control blocks.
#
# DM [= {0|1|2|3}]
#           Domain manager.
#
#           0           Suppress formatting of DM control blocks.
#           1           Format the wait queue.
#           2           Format the anchor block.
#           3           Format level-1 and level-2 data.
#
# DS [= {0|1|2|3}]
#           Dispatcher domain.
#
#           0           Suppress formatting of DS control blocks.
#           1           Format the dispatcher dump summary.
#           2           Format the anchor block.
#           3           Format level-1 and level-2 data.
#
# DU [= {0|2}]
#           Dump domain.
#
#           0           Suppress formatting of DU control blocks.
#           2           Format the DU anchor block.
#
# FCP [= {0|2}]
#           File control program.

```



```

#           0      Suppress formatting of the file control table.
#           2      Format the file control table.

#           ICP[={0|2}]
#           Interval control program.

#           0      Suppress formatting of the interval control elements (ICEs).
#           2      Format the ICEs.

#           IND[={0|1|2|3}]
#           Page number indexes for the formatted output.

#           0      Suppress formatting of the page number indexes.
#           1      Provide a control block index sorted by address.
#           2      Provide a control block index sorted by block name.
#           3      Format level-1 and level-2 data.

#           Note: IPCS does not produce page numbers if formatting directly to the
#           terminal.

#           JCP [={0|2}]
#           Journal control area.

#           0      Suppress formatting of the JCA.
#           2      Format the JCA.

#           KE[={0|1|2|3}]
#           CICS kernel.

#           0      Suppress formatting of the kernel control blocks.
#           1      Format the stack and a summary of tasks.
#           2      Format the anchor block.
#           3      Format level-1 and level-2 data.

#           LD[={0|1|2|3}]
#           Loader domain.

#           0      Suppress formatting of loader domain control blocks.
#           1      Format a program status and module summary.
#           2      Format the anchor block, the current program elements (CPEs), and
#           the active program elements (APEs)
#           3      Format level-1 and level-2 data.

#           LG[={0|1|2|3}]
#           Log manager domain.

#           0      Suppress formatting of log manager domain control blocks.
#           1      Format the log manager summary.
#           2      Format all log manager control blocks.
#           3      Format level-1 and level-2 data.

#           LM[={0|1|2|3}]
#           Lock manager domain.

#           0      Suppress formatting of lock manager domain control blocks.
#           1      Format the lock status and allocated locks summary.
#           2      Format the anchor block and quickcells.
#           3      Format level-1 and level-2 data.

```

```

#           ME[={0|2}]
#           Message domain.
#
#           0           Suppress formatting of the ME anchor block.
#           2           Format the anchor block.
#
#           MN[={0|1|2|3}]
#           Monitoring domain.
#
#           0           Suppress formatting of monitoring domain control blocks.
#           1           Format the monitoring status and monitoring dictionary summary.
#           2           Format the anchor block and monitoring control table.
#           3           Format level-1 and level-2 data.
#
#           MRO[={0|2}]
#           CICS multiregion operation.
#
#           0           Suppress formatting of all MRO control blocks.
#           2           Format MRO control blocks, APPC URDs, and any associated DWE
#           chains.
#
#           PA[={0|2}]
#           Parameter manager domain.
#
#           0           Suppress formatting of the PA anchor block.
#           2           Format the PA anchor block.
#
#           PCT[={0|2}]
#           Program control table.
#
#           0           Suppress formatting of the transaction definitions.
#           2           Format the transaction definitions.
#
#           PG[={0|1|2|3}]
#           Program manager domain.
#
#           0           Suppress formatting of program manager domain control blocks.
#           1           Format the program manager summary.
#           2           Format the program manager control blocks, including the anchor
#           block, the LLEs, the PGWEs, the PPTEs, the PLCBs, and the HTBs.
#           3           Format level-1 and level-2 data.
#
#           PR [={0|2}]
#           Partner resource management.
#
#           0           Suppress formatting of PR areas.
#           2           Format the PR static storage and the partner resource table.
#
#           RM [={0|2}]
#           Recovery manager domain.
#
#           0           Suppress formatting of RM control blocks.
#           2           Format RM control blocks.
#
#           SM[={0|1|2|3}]
#           Storage manager domain.
#
#           0           Suppress formatting of storage manager domain control blocks.

```

```

#           1       Format the dynamic storage areas (DSAs), task and domain storage
#           #       subpools, transaction areas (SMXs), suspend queue, and subspace
#           #       area (SUA) summaries.
#           2       Format the anchor block (SMA), subpool control areas (SCAs),
#           #       pagepool areas (PPAs), pagepool extent areas (PPXs), storage
#           #       manager transaction areas (SMXs), subspace areas (SUAs), suspend
#           #       queue elements (SQEs), page allocation maps (PAMs), DSA extent
#           #       descriptors (DXEs), and DSA extent getmain descriptors (DXGs).
#           3       Format level-1 and level-2 data.

#           SSA[={0|2}]
#           Static storage areas.

#           0       Suppress formatting of the static storage areas address list.
#           2       Format the static storage areas address list.

#           ST[={0|1|2|3}]
#           Statistics domain.

#           0       Suppress formatting of statistics domain control blocks.
#           1       Format statistics collection details.
#           2       Format the anchor block.
#           3       Format level-1 and level-2 data.

#           SZ[={0|1}]
#           Front end programming interface (FEPI).

#           0       Suppress formatting of the FEPI static area.
#           1       Format the FEPI static area. Nothing is printed unless you have
#           #       installed FEPI. See the CICS/ESA Front End Programming Interface
#           #       User's Guide for further information on this feature.

#           TCP[={0|1|2|3}]
#           Terminal control program.

#           0       Suppress formatting of TCP control blocks.
#           1       Format the terminal control summary.
#           2       Format the terminal control table, the terminal input/output areas, and
#           #       the AIDs.
#           3       Format level-1 and level-2 data.

#           TDP[={0|1|2|3}]
#           Transient data program.

#           0       Suppress formatting of transient data control blocks.
#           1       Format the summary of transient data destinations.
#           2       Format the transient data static storage areas and the multiple strings
#           #       control blocks (MRCBs).
#           3       Format level-1 and level-2 data.

#           TI[={0|1|2|3}]
#           Timer domain.

#           0       Suppress formatting of timer domain control blocks.
#           1       Format outstanding request details.
#           2       Format the anchor block.
#           3       Format level-1 and level-2 data.

```

```

#           TMP[={0|2}]
#           Table manager program.
#
#           0           Suppress formatting of table manager static storage and control
#           blocks.
#           2           Format table manager static storage and control blocks.
#
#           TR[={0|1|2|3}
#           Trace domain.
#
#           0           Suppress formatting of trace.
#           1           Format abbreviated trace.
#           2           Format full trace.
#           3           Format level-1 and level-2 data.
#
#           TRS[={<trace selectivity parameter(s)>}]
#           Trace entry selectivity.
#
#           This keyword is effective only if the TR keyword value is 1, 2, or 3.
#
#           The TRS component keyword allows you to exercise much the same selectivity
#           over the formatting and printing of trace entries written in a trace internal to a
#           system dump, as you may exercise over the formatting and printing of trace
#           entries in an auxiliary trace.
#
#           The trace selectivity parameter may be any valid trace selectivity parameter
#           available to &dfhtu. for the formatting of CICS auxiliary trace entries except the
#           parameters PAGESIZE, ABBREV, and FULL. You may, as with &dfhtu., select
#           any number of parameters from those available.
#
#           Note, however, that you must use angled brackets around the parameter, or
#           sequence of parameters, that you specify. The format and default values of
#           parameters used to select trace entries from an internal SDUMP trace, are the
#           same as those that apply when you use &dfhtu. to format auxiliary trace
#           entries.
#
#           TSP[={0|1|2|3}]
#           Temporary storage domain.
#
#           0           Suppress formatting of temporary storage control blocks.
#           1           Format a summary of temporary storage control blocks and temporary
#           storage control block checking.
#           2           Format temporary storage control blocks.
#           3           Format level-1 and level-2 data.
#
#           Apar PQ09198
#
#           Documentation for Apar PQ09198 added 07/07/98
#
#           The options for TSP may also be specified with angled brackets, for:
#
#           1           Summary
#           2           Format control blocks
#           3           Consistency checking of the TS buffers with the TS control blocks
#
#           One or more of these values may be specified between angled brackets; for
#           example, TSP=<1,2> to give summary and formatting of control blocks, without
#           consistency checking being produced.

```

```

#           UEH[={0|2}]
#           User exit handler.
#
#           0       Suppress formatting of control blocks.
#           2       Format control blocks.
#
#           US[={0|1|2|3}]
#           User domain.
#
#           0       Suppress formatting of user domain control blocks.
#           1       Format the user domain summary.
#           2       Format the control blocks.
#           3       Format level-1 and level-2 data.
#
#           XM[={0|1|2|3}]
#           Transaction manager.
#
#           0       Suppress formatting of transaction manager control blocks.
#           1       Format the domain summary, global state summary, transaction
#                   summary, transaction class summary, and MXT summary.
#           2       Format the control blocks including the transaction domain anchor
#                   block, transactions (TXn), and transaction class control blocks (TCL).
#           3       Format level-1 and level-2 data.
#
#           XRF[={0|2}]
#           Extended recovery facility.
#
#           0       Suppress formatting of control blocks.
#           2       Format control blocks.
#
#           XS[={0|1}]
#           Security domain.
#
#           0       Suppress formatting of anchor block and supervisor storage.
#           1       Format anchor block and supervisor storage.

```

Trace entry selection: The TRS component keyword allows you to exercise much the same choice over the formatting and printing of trace entries written in a trace internal to a system dump as you may exercise over the formatting and printing of trace entries in an auxiliary trace.

Note: The TRS keyword is effective only if the TR keyword value is 1, 2, or 3.

The trace selection parameters may be any valid trace selection parameters available to DFHTU410 for the formatting of CICS auxiliary trace entries, except the parameters PAGESIZE, ABBREV, and FULL. You may, as with DFHTU410, select any number of parameters from those available. (For descriptions of available parameters, see “The trace selection parameters” on page 164.)

Note: You must use angled brackets around the parameter, or sequence of parameters, that you specify. The format and default values of parameters used to select trace entries from an internal SDUMP trace, are the same as those that apply when you use DFHTU410 to format auxiliary trace entries.

The dump summary and error index

The CICS dump exit always produces a dump summary, even if you suppress all the component areas by specifying DEF=0 without component keywords. The dump summary always appears at the head of the dump, and contains the following:

- Dump identifier
- Dump code
- Date and time at which the SDUMP was taken
- Message text associated with the dump
- Symptom string
- Dump title
- Caller
- Address space ID.

Figure 41 gives an example of a dump summary.

```
=== DUMP SUMMARY
DUMPID: 1/0001
DUMPCODE: AP0001
DATE/TIME: 7/01/94 16:18:08 (LOCAL)
MESSAGE: DFHAP0001 applid AN ABEND (CODE 0C1/AKEA) HAS OCCURRED AT OFFSET
          X'00000076' IN MODULE DFHAPDM.
SYMPTOMS: PIDS/5685XX083 LVLS/320 MS/DFHAP0001 RIDS/DFHAPDM PTFS/ULnnnnn
          AB/S00C4 AB/UAKEA ADRS/00000076

TITLE: (NONE)
CALLER: (NONE)
ASID: X'001D'
```

Figure 41. Example dump summary

An error message index is produced if an error or information message is output while the CICS dump exit is formatting the dump data. The error message index is organized as follows:

- It is sorted by page number, giving the page numbers on which error or information messages have been output.
- There are separate indexes for the information and error messages.
- It contains a summary giving totals of messages.

Sample jobs to process a CICS SDUMP using the CICS dump exit

This section shows two sample jobs that you can use for processing CICS SDUMPs using IPCS. The first, in Figure 42 on page 193, is an example of how to create an IPCS dump directory; the second, in Figure 43 on page 194, is an example of a job that invokes IPCS from the TSO terminal monitor program to selectively print parts of a CICS dump. The latter specifies the CICS410 dump exit on the VERBEXIT subcommand, and identifies the areas of the CICS SDUMP that are to be printed.

```

//IPCSDIR JOB (accounting information),CLASS=A,MSGCLASS=A,
//          MSGLEVEL=(1,1),NOTIFY=userid
//*****
//*      This job creates and initializes a VSAM KSDS dataset for
//*      use as an IPCS dump directory.  There are two job steps:
//*
//*      1. CREDDIR creates a dump directory dataset for use by IPCS
//*
//*      2. INITDIR initializes the dataset.
//*****
//CREDDIR EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//AMSDUMP DD SYSOUT=*
//SYSIN DD *
DEFINE CLUSTER (NAME(CICS410.IPCSDIR)           -           1
                VOLUMES(volid)                 -           2
                CYLINDERS(2 1)                  -
                BUFFERSPACE(65536)              -
                KEYS(128 0))                    -
                DATA( NAME(CICS410.IPCSDIR.DATA) -           1
                CONTROLINTERVALSIZE(4096))      -
                INDEX(NAME(CICS410.IPCSDIR.INDEX)) -           1
/*
//INITDIR EXEC PGM=IKJEFT01,REGION=1500K
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
IPCSDIR 'CICS410.IPCSDIR'                       1
END
/*
//

```

Figure 42 (Part 1 of 2). Sample job to create an IPCS dump directory

Notes:

- 1** Change 'CICS410' to a high-level qualifier of your own choosing.
- 2** Specify the volume identifier (in place of 'volid') of whichever disk volume you intend using for the IPCS directory.

Figure 43 on page 194 is the sample formatting job that you can use after you have created the IPCS dump directory.

```

//IPCSDUMP JOB (accounting information),CLASS=A,MSGCLASS=A,
//          MSGLEVEL=(1,1),NOTIFY=userid
//*****
//*          This job formats a CICS SDUMP by invoking IPCS via
//*          the TSO terminal monitor program in a batch job.
//*****
//IPCSDUMP EXEC PGM=IKJEFT01,REGION=4096K
//STEPLIB DD DSN=CICS410.SDFHLINK,DISP=SHR
//          DD DSN=CICS410.SDFHPARM,DISP=SHR
//DFHSDUMP DD DSN=SYS1.DUMPnn,DISP=SHR
//DFHSDUMP DD SYSOUT=*
//IPCSDDIR DD DSN=CICS410.IPCSDIR,DISP=SHR
//IPCSDDIR DD DSN=SYS1.PARMLIB,DISP=SHR
//          DD DSN=CICS410.SDFHPARM,DISP=SHR
//IPCSTOC DD SYSOUT=*
//IPCSPRNT DD SYSOUT=*
//SYSPROC DD DSN=SYS1.SBLSCLI0,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//IPCSDUMP.SYSTSIN DD *
IPCS NOPARM
DROPDUMP DD(DFHSDUMP)
SETDEF DD(DFHSDUMP) NOPROBLEM NOCONFIRM NOTERMINAL PRINT LIST
VERBEXIT CICS410 'UPPERCASE,DEF=1,DLI=2,KE=3,TR=2'
END
/*

```

Figure 43. Sample job to format a CICS SDUMP using IPCS and the CICS dump exit

Notes:

1 The batch job in Figure 43 includes STEPLIB statements for:

- The CICS410.SDFHLINK library, to enable the VERBEXIT subcommand of IPCS to invoke the CICS-supplied IPCS dump exit routine DFHPD410
- The CICS410.SDFHPARM library, to enable the IPCS job to find the CICS-supplied DFHIPCSP member. (See on page 180.)

2 Specify the number of the dump data set being processed instead of the 'nn'.

3 The DD statement for DFHSDUMP is optional. It is required only in the event of a program check during the dump formatting, in which case a dump is written to DFHSDUMP. If you omit the DD statement, you get message IEC130I. We recommend that you always include the DD statement.

4 Change 'CICS410' to the high-level qualifier you defined for the IPCS directory.

5 You must ensure that the DFHIPCSP member can be found by your IPCS job. You can either copy the DFHIPCSP member into the SYS1.PARMLIB library (so that it is in the same default library as BLSCECT) or provide an IPCSPARM DD statement to specify the library containing the IPCS control tables, as shown in the example JCL. For information about making the DFHIPCSP member available, see "Specifying DFHIPCSP CICS exit control data needed" on page 180.

6 The DD statement for IPCSTOC is required if you want a table of contents produced for the various component areas of the formatted dump. If you ensure that it precedes the IPCSPRNT statement, then the table of contents appears at the head of the dump, otherwise it appears at the end.

7 You need only code this SYSPROC DD statement if you want to use any of the IPCS CLISTS from the SYS1.SBLSCLI0 library, or any other CLISTS. If you have

other libraries containing CLISTs, concatenate them with the IPCS library on the SYSPROC DD statement.

8 The DROPDUMP statement removes previous directory entries, enabling the directory to be reused without the need to delete and redefine it.

9 The VERBEXIT statement specifies the verb name CICS410 to process CICS/ESA 4.1 system dump data. This corresponds to the IPCS dump exit routine DFHPD410, as specified in the DFHIPCSP member in the CICS410.SDFHPARM library. If the VERBEXIT statement refers directly to the dump exit routine (for example, VERBEXIT DFHPD410 ...) you do not need to make the DFHIPCSP module available to the IPCS job (as in note **5**).

+
+
+

Chapter 13. Monitoring utility programs

CICS provides two programs for processing any CICS monitoring data that is written to system management facilities (SMF) data sets. These two programs are:

- DFHMNDUP** A utility program that generates a monitoring dictionary record, in a sequential data set, for use with monitoring data extracted from SMF data sets.
- DFH\$MOLS** A print program for CICS monitoring data. DFH\$MOLS is a sample program to show you how you can code your own monitoring utility program to print CICS monitoring data.

This chapter describes:

- The purpose of the DFHMNDUP program, with a sample job to run it
- The steps required to extract and print monitoring data using the DFH\$MOLS sample program, or a utility program of your own modelled on the DFH\$MOLS sample program.

The monitoring dictionary utility program, DFHMNDUP

When CICS monitoring is switched on, and you activate the monitoring performance class (MNPER=ON), CICS first writes a performance dictionary record to the current SMF data set, and then begins to write the monitoring performance data records. A new dictionary record, which always precedes the monitoring data it relates to, is written whenever you:

- Start CICS with the performance class active, and CICS monitoring on
- Change the status of the monitoring performance class from inactive to active, with CICS monitoring on. If monitoring is off and the monitoring performance class is switched from inactive to active, a dictionary record is scheduled from the next time monitoring is activated.

Any monitoring utility program that processes performance data must read the dictionary record that relates to the data being processed before attempting to analyze the data. However, if SMF switches data sets during the period when CICS monitoring is writing performance data, CICS does not write a new dictionary record, and therefore a CICS performance dictionary record is not the first monitoring performance record on the new SMF data set. The DFHMNDUP program provides a solution to the problem posed by SMF data sets that do not contain a dictionary record.

The performance dictionary record

A performance dictionary record holds specific information about each data field in a performance data record. It derives its information from predefined CICS fields, and from any user-defined fields in the MCT specified for the CICS run. For programming information about the performance dictionary and data records, see the *CICS/ESA Customization Guide*. For information about the MCT definitions, see the *CICS/ESA Resource Definition Guide* manual.

To enable you to process SMF data sets that contain performance data records but not a dictionary record, DFHMNDUP writes a dictionary record to a sequential data

set. The dictionary record is written to a data set specified on a DD statement with a ddname of SYSUT4. You must put this data set in front of any data set(s) you are processing and which contain performance data. You provide control information for the DFHMNDUP program in the SYSIN data set so that it can generate the correct dictionary record for the performance data you are processing. The values that are specified are used to construct the fields in the SMF Header and Product section. The relationship of DFHMNDUP SYSIN parameters to SMF fields is shown in Table 15.

SYSIN PARM	SMF Dsect field	Meaning
DATE	SMFMNDTE	Date record moved
GAPPLID	SMFMNPRN	Product name (Generic APPLID)
JOBDATE	SMFMNRSD	Job execution date
JOBNAME	SMFMNJBN	Jobname of CICS job
JOBTIME	SMFMNRST	Job execution time
SAPPLID	SMFMNSPN	Specific APPLID
SYSID	SMFMNSID	System identification
TIME	SMFMNTME	Time record moved
UPPERCASE	n/a	Uppercase output
USERID	SMFMNUIF	User identification

You specify control information for the DFHMNDUP program on the following parameters:

DATE=yyddd or DATE=yyyyddd

Code this with the Julian date to be included in the dictionary record, where:

yy represents the year of the twentieth century (for example 96 for 1996)

yyyy represents the year (For the twenty-first century the year must be represented by yyyy. If yy is coded the twentieth century is assumed)

ddd represents the day, in the range 1 through 366

For example, the date 95354 represents the 20th December 1995 and the date 2005354 represents the 20th December 2005. If you do not specify a date, the current date is used.

GAPPLID=name Code this with the generic VTAM APPLID of the CICS region for which you are analyzing performance data.

JOBDATE=yyddd or JOBDATE=yyyyddd

Code this with the MVS job date (in Julian date format) to be included in the dictionary record.

yy represents the year of the twentieth century (for example 96 for 1996).

+ **yyyy** represents the year. (For the twenty-first century
+ the year must be represented by yyyy. If yy is
+ coded the twentieth century is assumed).

+ **ddd** represents the day, in the range 1 through 366.

+ For example, the date 95354 represents the 20th December
+ 1995 and the date 2005354 represents the 20th December
+ 2005.

| If you do not specify a date, the current date is used.

|

| **JOBNAME=xxxxxxx** Code this with an MVS job name for the CICS region to be
| included in the dictionary record.

|

| **JOBTIME=hhmmss** Code this with a time stamp, as six numeric characters, for
| MVS job to be included in the dictionary record.

| **hh** the number of hours, in the range 00 through 24.

| **mm**
| the number of minutes, in the range 00 through 59.

| **ss** the number of seconds, in the range 00 through 59.

| If you do not specify a time, the current time is used.

|

| **MCT=xx** Code this with the suffix of the monitoring control table (MCT)
| used in the CICS run for which you are analyzing performance
| data. If your CICS region ran with the system initialization
| parameter MCT=NO (which results in a default MCT
| dynamically created by CICS monitoring domain) you should
| specify MCT=NO for DFHMNDUP also. Alternatively, you can
| indicate that your CICS used a default MCT by specifying
| 'MCT=' or 'MCT=,' where appropriate (see below).

Note: Using DFHMNDUP to generate a monitoring dictionary
record, means that you must use two-character suffixes
for MCTs. If you create an MCT with only a
single-character suffix, it is accepted by CICS but
rejected by DFHMNDUP.

|

| **SAPPLID=name** Code this with the specific VTAM APPLID of the CICS region
| for which you are analyzing performance data. If you omit this
| parameter, the value you specify on the GAPPLID parameter is
| taken as the specific APPLID also.

|

| **SYSID=xxxx** Code this with the system identifier of the MVS system that
| owns the SMF data sets.

|

| **TIME=hhmmss** Code this with a time stamp for the dictionary record. If you do
| not specify a time, the current time is used.

|

| **UPPERCASE** Code this parameter if you want the statistics output in
| uppercase only. If you want output in mixed case (the default),
| do not code this parameter.

USERID=xxxxxxx

Code this with eight alphanumeric characters that represent the user identification of the MVS job to be included in the dictionary record. The user identification value xxxxxxxx, must correspond to any values that you have set up in your MVS IEFUSI exit, but does not have to be a real userid. For information on the MVS job step initiation exit IEFUSI, see the *MVS/ESA Installation Exits* manual.

You can enter each parameter on a separate line, with the parameter keyword starting in column one. Alternatively, you can enter all of the parameters on a single line, starting in column one, with each parameter separated by a comma. If your CICS used a default MCT, you can enter the MCT parameter as 'MCT=NO', 'MCT=', or 'MCT=,'.

For example, you can use the following three methods to specify the same control information for the DFHMNDUP program:

- (MCT=NO)

```
//SYSIN DD *
MCT=NO
SYSID=MVSA
GAPPLID=DBDCCICS
SAPPLID=DBDCCIC1
DATE=89256
TIME=000001
/*
```

- (MCT=)

```
//SYSIN DD *
MCT=
SYSID=MVSA
GAPPLID=DBDCCICS
SAPPLID=DBDCCIC1
DATE=89256
TIME=000001
/*
```

- (MCT=,)

```
//SYSIN DD *
MCT=,SYSID=MVSA,GAPPLID=DBDCCICS,SAPPLID=DBDCCIC1,DATE=89256,TIME=000001
/*
```

Sample job illustrating the use of DFHMNDUP

The sample job in Figure 44 on page 201 shows how you can use the DFHMNDUP program in conjunction with the DFH\$MOLS program to print monitoring performance data.

```

//SMFMNDUP JOB (accounting information),CLASS=A,
//      MSGCLASS=A,USER=userid,PASSWORD=password,NOTIFY=userid
//*****
/* Step 1 - Create new dictionary record and output to SYSUT4 *
//*****
//MNDUP EXEC PGM=DFHMNDUP
//STEPLIB DD DSN=CICS410.SDFHLOAD,DISP=SHR
//      DD DSN=mct.table.loadlib,DISP=SHR
//SYSUT4 DD DSN=CICS410.applid.MNDUPREC,DISP=(NEW,CATLG),
//      UNIT=SYSDA,SPACE=(TRK,(1,1))
//SYSPRINT DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//SYSIN DD *
MCT=NO
SYSID=MVSA
GAPPLID=DBDCCICS
SAPPLID=DBDCCIC1
DATE=91205
TIME=000100
/*
//*****
/* Step 2 - Unload the SMF data set containing CICS data *
//*****
//SMFDUMP EXEC PGM=IFASMFDP
//INDD1 DD DSN=SYS1.MANx,DISP=SHR,AMP=('BUFSP=65536')
//OUTDD1 DD DSN=SYS1.SMFDMPnn,DISP=(NEW,CATLG),
//      SPACE=(CYL,(10,2)),UNIT=SYSDA
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
          INDD(INDD1,OPTIONS(DUMP))
          OUTDD(OUTDD1,TYPE(110))
/*
//*****
/* Step 3 - Run DFH$MOLS to print the CICS monitoring data, *
/*      using the new dictionary record from step 1 *
//*****
//PRNTMND EXEC PGM=DFH$MOLS
//STEPLIB DD DSN=CICS410.SDFHLOAD,DISP=SHR
//INPUT DD DSN=CICS410.applid.MNDUPREC,DISP=OLD
//      DD DSN=SYS1.SMFDMPnn,DISP=OLD
//SYSPRINT DD SYSOUT=A
//SYSABEND DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//SYSIN DD *
(DFH$MOLS control statements - see page 205)

```

Figure 44. Sample job stream to run DFHMNDUP

Notes:

1 In addition to the CICS library containing the DFHMNDUP program, the STEPLIB library concatenation must also include the library that contains any monitoring control table (MCT) that you specify on the MCT parameter.

2 You may decide to keep a permanent data set, one for each CICS region, to hold the dictionary record. Specify the DISP parameter according to whether the data set already exists, or a new one is to be created and cataloged.

3 Specify the name of the SMF data set that you want to dump, where “x” is the installation-defined suffix in the range A to Z, or 1 to 9. You can reduce the time to unload the SMF data set by including an AMP parameter with a suitable buffer size. For further information about unloading SMF data sets, see the *MVS/ESA System Management Facilities* manual.

4 If you decide to dump to a permanent SMF dump data set, specify the DISP parameter according to whether the data set already exists, or a new one is to be created and cataloged. The naming convention shown here suggests the use of a suffixed name (SMFDMPxx) for the low-level qualifier, where xx is a two-character suffix.

5 You must put the dictionary data set in front of the dumped SMF data set. If the first monitoring performance record in the SMF data set is not a dictionary record, the dictionary record created by DFHMNDUP is used. However, if the first monitoring performance record in the SMF data set *is* a dictionary record, it is used instead of the dictionary record created by the DFHMNDUP program. The DFH\$MOLS sample uses the last dictionary record read and disregards any previous record.

The sample monitoring data print program, DFH\$MOLS

This section describes how you can unload monitoring data from SMF data sets, for one or more CICS regions, and print the data using the CICS sample utility program, DFH\$MOLS. This utility is a CICS-supplied batch program which you can modify or adapt to your own purposes. The job steps involved in processing CICS monitoring data are:

1. Unload the SMF data set(s) so that the SMF data is available for processing by a CICS utility. For information about unloading SMF data sets, see the *MVS/ESA System Management Facilities* manual.
2. Run the DFH\$MOLS program to print monitoring records, which you can optionally select and sort by means of control statements.

Overview of the DFH\$MOLS program

The DFH\$MOLS program is a data reduction program designed to produce reports from the data collected by the CICS monitoring domain (MN), and written to SMF data sets. You run the DFH\$MOLS program in a batch region to process any CICS SMF type 110 monitoring records that are present in an unloaded SMF data set, which you can write to either a temporary or cataloged data set. You can determine the scope of the report(s) by supplying control statements in the SYSIN data set.

You can specify a sort option for the selected data. The DFH\$MOLS program sorts the data by means of a link to the MVS sort program, DFSORT, passing parameters to the sort, and using the sort exits E15 and E35. You can use any standard sort utility provided it has these E15 and E35 exits. For further information about the DFSORT program, see the *DFSORT Application Programming Guide*.

The program reads, formats, and prints the CICS monitoring data, which is packaged in the following format:

[SMF HEADER].[SMF PRODUCT SECTION].[CICS DATA SECTION]

The CICS data section in a monitoring record is one of the following:

1. A dictionary data section, consisting of a sequence of dictionary entries
2. An exception data section, consisting of a single exception record
3. A performance data section, consisting of a sequence of field connectors followed by one or more performance records.

For programming information about the structure of CICS SMF type 110, and how the monitoring data is packaged within the SMF records, see the *CICS/ESA Customization Guide*. The DFH\$MOLS program removes the SMF data and moves individual CICS monitoring data fields to the print area. If you want to analyze the data using your own routines, this is the point at which you can link to a user-written analysis program.

Monitoring dictionary records

The DFH\$MOLS program requires a monitoring dictionary record to process monitoring performance data. When it locates a dictionary record, it builds an in-store dictionary and processes the subsequent (if any) performance data using this dictionary. Whenever it reads a new dictionary record, the current dictionary is released and a new in-store dictionary is built. The dictionary record must appear before any related performance data, otherwise the DFH\$MOLS program abends. Note that monitoring exception records do not require a dictionary and so they can precede the first dictionary record and still be successfully processed. For more information about monitoring dictionary records, see “The performance dictionary record” on page 197.

Volume of output

The DFH\$MOLS program prints about one page per task, so take care to specify only those items that you need using the DFH\$MOLS program control statements. For details of the selection options, see “Control statements of the DFH\$MOLS program” on page 205.

Sample job stream for the DFH\$MOLS programs

Figure 45 shows a sample job with the SMF unload step as well as the monitoring report step.

```
//MONPRNT JOB (accounting information),CLASS=A,
//          MSGCLASS=A,MSGLEVEL=(1,1)
//SMFUNLD  EXEC PGM=IFASMFDP
//INDD1    DD DSN=SYS1.MANx,DISP=SHR,AMP=('BUFSP=65536')
//OUTDD1   DD DSN=&&TEMP,DISP=(NEW,PASS),SPACE=(CYL,(2,1)),
//          UNIT=SYSDA
//SYSPRINT DD SYSOUT=A
//SYSIN    DD *
           INDD(INDD1,OPTIONS(DUMP))
           OUTDD(OUTDD1,TYPE(110))
/*
/**
/** Select CICS/ESA monitoring records
/**
//RECESEL  EXEC PGM=DFHJUP
//STEPLIB DD DSN=CICS410.SDFHLOAD,DISP=SHR
//SYSUT1   DD DSN=&&TEMP,DISP=(OLD,PASS)
//SYSUT4   DD DSN=&&TEMP2,DISP=(NEW,PASS),SPACE=(CYL,(5,2)),
//          UNIT=SYSDA
//SYSPRINT DD SYSOUT=A
//SYSIN    DD *
* CICS/ESA Version 3 and later records
OPTION    COPY OFFSET=25,FLDTYP=X,VALUE=0002,FLDLLEN=2,COND=E
END
/**
//PRNT     EXEC PGM=DFH$MOLS
//STEPLIB DD DSN=CICS410.SDFHLOAD,DISP=SHR
//INPUT    DD DSN=&&TEMP,DISP=(OLD,DELETE),UNIT=SYSDA
//SORTWK01 DD SPACE=(CYL,(5,1)),UNIT=SYSDA
//SORTWK02 DD SPACE=(CYL,(5,1)),UNIT=SYSDA
//SORTWK03 DD SPACE=(CYL,(5,1)),UNIT=SYSDA
//SORTWK04 DD SPACE=(CYL,(5,1)),UNIT=SYSDA
//SORTWK05 DD SPACE=(CYL,(5,1)),UNIT=SYSDA
//SORTDIAG DD SYSOUT=A
//SYSOUT   DD SYSOUT=A
//SYSPRINT DD SYSOUT=A
//SYSABEND DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//SYSIN    DD *
PRINT PER
:
Control statements for data selection
:
SORT
/**
```

Figure 45. Sample job to unload and process CICS data from SMF data sets

Notes

1 Specify the last character of the data set name (in place of 'x') for the SMF data set you are unloading. For information about unloading multiple SMF data sets, see the notes to the sample DFHSTUP statistics job on page 154.

2 If you want to keep the unloaded data set, change the DSN and DISP parameters appropriately.

3 The SMF dataset may contain any type of SMF record, but in this example we are unloading CICS type 110 records only. Although this may include CICS statistics records, and any CICS user journal records written by SMF, the DFH\$MOLS program ignores them and process monitoring data only; they are

identified by a record sub-type '01'. Specify TYPE(0:255) to unload all types of SMF record.

4 If CICS/MVS Version 2 and CICS/ESA Version 3 or later SMF records have been written to the same SMF data set, you should use the DFHJUP utility program to select either the CICS/ESA or CICS/MVS records and copy them into a separate data set. These records are identified by offset 25 in the SMF records. A value of X'0002' identifies a CICS/ESA record, and a value of X'0000' identifies CICS/MVS record.

5 If you have generated your own version of the DFH\$MOLS program and stored it in a different library from the CICS-supplied version, change the STEPLIB statement accordingly. On the STEPLIB statement you should specify the library that contains the version of the DFH\$MOLS program for the CICS release of SMF records to be formatted. (For example, CICS410.SDFHLOAD for CICS/ESA 4.1 SMF records, or CICS212.LOADLIB for CICS/MVS 2.1.2 records.)

6 These sort work files are needed only if you specify the sort option. SORT is recommended to ensure that data is correctly processed in chronological order.

7 Specify the control statements for data selection and other options in SYSIN. For details of the control statements for the DFH\$MOLS program, see "Control statements of the DFH\$MOLS program."

Control statements of the DFH\$MOLS program

You control the processing that the DFH\$MOLS program performs by specifying the following control statements (for more details about these statements, see page 206):

* [user comments]

BREAK Use this statement to group SELECT/IGNORE statements into logical groups.

CONTROL STOPAFT=nnnnnnnn

DATE START=mm/dd/yy or mm/dd/yyyy,STOP=mm/dd/yy or mm/dd/yyyy

For the twenty-first century the year must be represented by yyyy. If yy is coded the twentieth century is assumed, for example 96 for 1996.

IGNORE APPLID=xxxxxxx[,yyyyyyy,...]

TERMID=xxxx[,yyyy,...]

TRANID=xxxx[,yyyy,...]

USERID=xxxxxxx[,yyyyyyy,...]

You can use any of these IGNORE options in conjunction with SELECT statements to form SELECT/IGNORE groups (see the BREAK control statement).

PRINT EXC,PER,ALL

SELECT APPLID=xxxxxxx[,yyyyyyy,...]
TERMINID=xxxx[,yyyy,...]
TRANID=xxxx[,yyyy,...]
USERID=xxxxxxx[,yyyyyyy,...]

You can use any of these SELECT options in conjunction with IGNORE statements to form SELECT/IGNORE groups (see the BREAK control statement).

SORT Use this statement to sort the input monitoring data before processing.

TIME START=hh.mm.ss,STOP=hh.mm.ss

TIMEOFF Use this statement to suppress testing for data output chronologically.

Rules for coding control statements

Control statements are free format, each statement having an operation keyword followed by a parameter that can have one or more operands. The DFH\$MOLS program pads operands of fewer characters than the permitted maximum with trailing blanks. Multiple operands are separated by commas, but if you specify only a single operand, a blank indicates the end of the operand. You can code any characters (except comma and blank) in the operands.

There are no continuation statements; you can specify multiple occurrences of the same control statement keyword, eliminating the need for continuations.

The DFH\$MOLS program prints each control statement before analyzing it. If the DFH\$MOLS program detects an error, it is associated with the last statement printed. Control statement errors are followed by an abend U101, without a dump.

If you do not specify any control statements, the DFH\$MOLS program produces a default listing of the monitoring data, using default values.

Control statement descriptions

* (asterisk)

An asterisk appearing in column 1 means that the statement is a comment only and has no effect on processing. It is printed without any further analysis.

BREAK

Code this control statement to end a SELECT/IGNORE group. The BREAK control statement logically terminates the previous (un-grouped) SELECT/IGNORE statements, and the DFH\$MOLS program forms them into a logical SELECT/IGNORE group. A BREAK statement has no meaning unless preceded by one or more SELECT/IGNORE statements. You can form multiple SELECT/IGNORE groups by including BREAK statements at the appropriate points.

The main intent of BREAK is to allow SELECT statements to be processed as a logical OR function instead of as a logical AND function if they are in the same group.

If you do not specify a BREAK statement after the last SELECT/IGNORE statement, the DFH\$MOLS program assumes one by default. This means that there is always at least one SELECT/IGNORE group, however few SELECT or IGNORE statements you specify.

If you specify only one SELECT/IGNORE group, either implicitly or by including a BREAK statement, the SELECT/IGNORE logic is the same as described for the individual SELECT/IGNORE statements.

The DFH\$MOLS program processes multiple SELECT/IGNORE groups using the following rules in the order listed:

1. Processing starts with the first group.
 2. The DFH\$MOLS program processes the SELECT/IGNORE groups in the order in which you specify them in SYSIN.
 3. APPLID, USERID, TERMID, and TRANID parameters may all be included in the same run.
 4. You cannot specify SELECT and IGNORE for the same parameter in the same SELECT/IGNORE group. For example, SELECT TERMID and IGNORE TERMID is invalid.
 5. If you specify SELECT for more than one parameter in a SELECT/IGNORE group, the SELECT statements form a logical AND function.
 6. If you specify IGNORE for more than one parameter in a SELECT/IGNORE group, the IGNORE statements form a logical OR function.
 7. The DFH\$MOLS program processes *all* SELECT statements in a group before any IGNORE statements in the same group.
 8. If a data record satisfies *all* of the SELECT statements in a group, it is selected (but subject to any following IGNORE statements).
 9. If a selected record (either by default in the absence of any SELECT statements, or explicitly because it satisfies selection criteria) also satisfies an IGNORE test, the record is excluded.
 10. If a record is not included or excluded after all of the SELECT/IGNORE statements in a group have been processed, it is processed by the next group.
 11. If a record is not specifically included or excluded after all of the SELECT/IGNORE groups have been processed, one of the following events occurs:
 - If you do not specify a group with IGNORE statements only, to specifically exclude the record, it is excluded by default.
 - If you specify one or more groups with IGNORE statements only, to specifically exclude *other* records, the record is included.
- Note:** You can specify one or more groups with IGNORE statements only to specifically exclude records. However, any record not included or excluded, after all the SELECT/IGNORE and IGNORE-only groups, is included.

Examples:

The following control statements select records for transaction id TSK1 which were entered from terminal id T040:

```
SELECT TRANID=TSK1
SELECT TERMID=T040
```

The following control statements select records for all records for transaction id TSK1, and all records from terminal id T040. The BREAK statement effectively creates two SELECT/IGNORE groups, and any record satisfying group 1 (the transaction id is TSK1) or group 2 (the terminal id is T040) is selected:

```
SELECT TRANID=TSK1
BREAK
SELECT TERMID=T040
```

The following control statements select records for transaction ids TSK1 and TSK2, but excluding those that were entered from terminal id T040:

```
SELECT TRANID=TSK1,TSK2
IGNORE TERMID=T040
```

The following control statements select all records for transaction id TSK1 (SELECT group 1) and all records for transaction id TSK2 but exclude those entered from terminal id T040 (SELECT/IGNORE group 2):

```
SELECT TRANID=TSK1
BREAK
SELECT TRANID=TSK2
IGNORE TERMID=T040
```

If you also have records for terminal ids T050 (for transaction ids TSK1 and TSK3) and T060 (for transaction id TSK3 only), you can use the following IGNORE-only group to exclude all records entered from terminal id T050:

```
IGNORE TERMID=T050
```

In this case, records for terminal id T060 are included, because you have not specifically excluded them.

To exclude the records from terminal ids T050 and T060, you can do one of the following:

- Do not specify any IGNORE-only groups; the records for terminal ids T050 and T060 are excluded by default.
- Specify one or more IGNORE-only groups, to specifically exclude records from terminal ids T050 and T060, for example:

```
IGNORE TERMID=T050,T060
```

In this case, if you later add another terminal, its records are included unless you specify the terminal id in an IGNORE-only group.

CONTROL STOPAFT=nnnnnnnn

Specify the CONTROL statement with its associated STOPAFT parameter to control the number of records you want to process.

STOPAFT=nnnnnnnn

Specify this parameter to limit the number of SMF type 110 records you want the DFH\$MOLS program to process. The DFH\$MOLS program terminates after processing the number of SMF 110 records specified by nnnnnnnn.

DATE

Specify the start and stop dates on this statement which, in conjunction with the TIME statement (if specified), enables you to select records for a particular period only. (See also the TIME control statement.)

START=start-date

- + Specify the date of the beginning of the period for which you want records processed, in the form mm/dd/yy or mm/dd/yyyy.
- + Start dates in the twenty-first century must use the form mm/dd/yyyy.

STOP=stop-date

- + Specify the date of the end of the period for which you want records processed, in the form mm/dd/yy or mm/dd/yyyy.
- + Stop dates in the twenty-first century must use the form mm/dd/yyyy.

You do not have to specify both START and STOP; you can specify START without STOP, and STOP without START. If you omit the DATE statement, records for all dates present in the input file are processed. You can specify only one DATE statement (and associated TIME statement) in SYSIN.

IGNORE [APPLID|TERMINID|TRANID|USERID]

Code an IGNORE control statement to exclude all records from a CICS region that has the specified generic APPLID, or all records that have a specified transaction, terminal, or user identifier.

APPLID=xxxxxxx[,yyyyyyy,..]

Specify one or more generic APPLIDs to exclude monitoring data from a CICS region, or regions.

TERMINID=xxxx[,yyyy,..]

Specify one or more terminal identifiers to exclude monitoring data associated with these terminals.

TRANID=xxxx[,yyyy,..]

Specify one or more transaction identifiers to exclude monitoring data for these transactions.

USERID=xxxxxxx[,yyyyyyy,..]

Specify one or more user identifiers to exclude monitoring data for transactions submitted by these users.

You can specify each of the APPLID, TERMINID, TRANID, and USERID parameters in the same SELECT/IGNORE GROUP, but you cannot specify an IGNORE and SELECT for the same type of parameter. For example, you *can* specify SELECT APPLID= and IGNORE TERMINID=, but you *cannot* specify SELECT APPLID= and IGNORE APPLID=.

The DFH\$MOLS program pads, with trailing blanks, operands that have less characters than the permitted maximum. You cannot continue control statements on another line, but the program logically chains multiple control statements of the same keyword in the same IGNORE group (see the BREAK control statement). If you specify IGNORE for more than one parameter, those IGNORE statements form a logical OR function.

Examples:

If you specify:

```
IGNORE TRANID=CEMT
IGNORE USERID=OP7
```

the program excludes all records for transaction CEMT (regardless of user ID), and exclude all records containing userid OP7 (regardless of transaction ID). It includes all other records.

If you specify:

```
SELECT TRANID=CEMT
IGNORE TERMID=TRM3
```

the program includes only records for transaction CEMT, except for those from terminal TRM3.

PRINT {ALL|EXC|PER|EXC,PER}

Specify on this statement the type of monitoring data record that you want printed.

- ALL** lists all of the monitoring SMF type 110 records that are selected by any other control statement options. This is the default if you omit the PRINT statement.
- EXC** lists only the monitoring exception class records that are selected by any other control statement options.
- PER** lists only the monitoring performance class records that are selected by any other control statement options.

Note that the SMF headers, SMF product sections, and CICS dictionary records are always printed.

SELECT [APPLID|TERMID|TRANID|USERID]

The SELECT control statement includes all records of the specified generic APPLIDs, transaction, terminal, or user identifiers.

APPLID=xxxxxxx[,yyyyyyy,..]

Specify one or more generic APPLIDs to include monitoring data from the CICS regions identified by these APPLIDs.

TERMID=xxx[,yyy,..]

Specify one or more terminal identifiers to include monitoring data associated with these terminals.

TRANID=xxx[,yyy,..]

Specify one or more transaction identifiers to include monitoring data for these transactions.

USERID=xxxxxxx[,yyyyyyy,..]

Specify one or more user identifiers to include monitoring data for transactions submitted by these users.

You can specify each of the APPLID, TERMID, TRANID, and USERID parameters in the same SELECT/IGNORE GROUP, but you cannot specify IGNORE and SELECT for the same type of parameter. For example, you **can** specify SELECT APPLID= and IGNORE TERMID=, but you **cannot** specify SELECT APPLID= and IGNORE APPLID=.

You cannot continue control statements on another line, but the program logically chains multiple control statements of the same keyword in the same SELECT group. (See the BREAK control statement for details of how to terminate a SELECT/IGNORE group.) If you specify SELECT for more than one parameter, those SELECT statements form a logical AND function.

Examples:

If you specify:

```
SELECT TERMID=TRM3
SELECT TRANID=CEMT
```

the program includes only records with a transaction identifier of CEMT **and** with a terminal identifier of TRM3. It does not include any other records.

If you specify:

```
SELECT APPLID=DBDCCICS
SELECT TRANID=CEMT
IGNORE TERMID=TRM3
```

the program includes only those records that are from the CICS region with the generic APPLID DBDCCICS, and are for transaction CEMT, but do **not** have the terminal identifier TRM3.

SORT

Specify SORT to request a sort of the input monitoring data before the records are processed for output. If you specify a sort, the DFH\$MOLS program links to the standard MVS SORT program, DFSORT. (For information about the standard sort used, see the *DFSORT Application Programming Guide*.)

The SORT utility sorts the monitoring data into the following sequence:

```
Generic APPLID at position 47
SMF record sub-type at position 23
SMF record date at position 11
SMF record time at position 7.
```

These sort fields are built into the DFH\$MOLS program, and therefore the SORT statement does not require any parameters. However, if you want to perform a stand-alone sort before running the sample utility, you should use the following SORT statements:

```
SORT FIELDS=(47,8,CH,A,23,2,BI,A,11,4,PD,A,7,4,BI,A),EQUALS
RECORD TYPE=V
```

Note: SORT is the recommended option when you are processing data from multiple SMF data sets, and must be used when processing data for multiple CICS regions.

TIME

Specify the start and stop times on this statement which, in conjunction with the DATE statement (if specified), enables you to select records for a particular SMF time period only. (The time stamp against which the DFH\$MOLS program compares is the SMF time in the SMF header, not the time in individual performance records. This means that the program may select performance records for times that may be a few minutes outside the specified period because of the way they are buffered for writing to SMF.)

Note: A TIME statement without a DATE statement causes The DFH\$MOLS program to select data for the specified time period for **all** dates present in the input data set.

START=start-time

Specify the start time of the period for which you want records processed, in the form hh.mm.ss or hhmmss. A start time is optional,

and if omitted the report includes all records for the start date, irrespective of time.

STOP=stop-time

Specify the end time of the period for which you want records processed, in the form hh.mm.ss or hhmss. An end time is optional, and if omitted the report includes all records for the stop-date, irrespective of time.

You do not have to specify both START and STOP; you can specify START without STOP, and STOP without START. You can specify only one TIME statement (and associated DATE statement) in SYSIN.

TIMEOFF

Specify the TIMEOFF statement to suppress testing for data being out of chronological sequence.

By default, the DFH\$MOLS program checks the date/time sequence of the data to prevent incorrect processing caused by data being out of sequence. This can occur if you do not specify the SORT option of the DFH\$MOLS program, and one of the following is true:

- The input data is from a data set incorrectly sorted prior to running the DFH\$MOLS program.
- The unsorted data is from multiple SMF data sets that are not concatenated in time ordered sequence, or are unloaded in the wrong sequence.

The default sequence check causes the DFH\$MOLS program to terminate if data is out of sequence. You are recommended to omit the TIMEOFF statement, unless you have some reason for suspecting that the data is out of chronological sequence, and want to obtain a full listing of all the data.

Abend codes and error messages for the DFH\$MOLS program

The DFH\$MOLS program can fail with one of the following abend codes or error messages:

101 INVALID, DUPLICATE, OR MISSING CONTROL CARD INFORMATION

Explanation: The DFH\$MOLS control statement scan routines produce this message. It occurs immediately after processing an invalid control card, and the last control statement printed contains the error. Some of the causes are:

- Misspelled control card operation
- Misspelled control card operand
- Invalid operand delimiter
- Operand length too long.

The message is followed by a U101 abend, without a dump.

102 INCOMPATIBLE CONTROL CARD OPTIONS REQUESTED

Explanation: Some of the selected control statement options are in conflict.

The error is followed by a U102 abend without a dump.

103 SORT ENDED WITH NON-ZERO RET CODE; REPORT IS TERMINATED

Explanation: The MVS SORT utility has returned an error code after being invoked. This may be due to missing or incorrect job control statements. The SORT utility writes a message to SYSOUT defining the error.

The message is followed by an MVS abend U103, with a dump.

104 NO DATA FOR THIS CICS DATA SECTION; REPORT IS TERMINATED

Explanation: The DFH\$MOLS program has detected a CICS 110 monitoring record that does not contain any data rows within the data section. The DFH\$MOLS program detects this by validating the contents for field SMFMNDRN in the SMF product section.

The message is followed by an MVS abend U104 with a dump.

105 UNKNOWN CLASS IN CICS DATA SECTION; REPORT IS TERMINATED

Explanation: The DFH\$MOLS program has detected a CICS 110 monitoring record that does not contain a valid data section. The DFH\$MOLS program detects this by validating the contents for field SMFMNCL in the SMF product section, which should indicate that the data section contains either dictionary data, performance data, or exception data.

The message is followed by an MVS abend U105 with a dump.

106 NO VALID MONITORING RECORDS WERE READ; REPORT IS TERMINATED

Explanation: There is no valid CICS monitoring data in the input data set.

The message is followed by an MVS abend U106, without a dump.

107 NO MONITORING RECORDS WERE SELECTED FOR PROCESSING; REPORT IS TERMINATED

Explanation: Valid CICS monitoring records have been read, but no records are eligible for processing because of SELECT/IGNORE control statements.

This message is followed by an MVS abend U107 without a dump.

108 X'nnnn' - CONNECTOR NOT DEFINED; REPORT IS TERMINATED

Explanation: A CICS 110 monitoring performance class record has a connector number that is not in the current dictionary.

This message is followed by an MVS abend U108 with a dump.

109 NO DICTIONARY ESTABLISHED; REPORT IS TERMINATED

Explanation: The first CICS 110 monitoring performance class record read by the DFH\$MOLS program has not been preceded by the corresponding dictionary record. This abend is issued if a dictionary record does not exist or if the dictionary record, created by DFHMNDUP, has a DATE (SMF record date: SMFMNDTE) and TIME (SMF record time: SMFMNTME) later than the DATE and TIME of the first monitoring record read. DFH\$MOLS sorts the monitoring data into the sequence described below to enable it to process the data correctly.

```

#           Generic APPLID at position 47
#           SMF record sub-type at position 23
#           SMF record date at position 11
#           SMF record time at position 7.
```

The DFH\$MOLS program must establish a dictionary record for each APPLID before it can correctly analyze any associated monitoring performance class records. (See "The monitoring dictionary utility program, DFHMNDUP" on page 197 and "Monitoring dictionary records" on page 203.)

```

#
#
#
#
#
#
#
#
#
#
#
```

This message is followed by an MVS abend U109 without a dump.

110 NO DICTIONARY ESTABLISHED FOR NEW APPLID xxxxxxxx; REPORT IS TERMINATED

Explanation: The DFH\$MOLS program has detected a change of APPLID in CICS 110 monitoring performance class data before it has read the corresponding dictionary record for the new APPLID. The DFH\$MOLS program must establish the new dictionary record for the new APPLID before it can correctly analyze any associated monitoring performance class records. (See "Monitoring dictionary records" on page 203.)

This message is followed by an MVS abend U110 without a dump.

111 DATE/TIME STEPDOWN ERROR ON INPUT; REPORT IS TERMINATED

Explanation: The DFH\$MOLS program has detected that the data is out of sequence. CICS writes monitoring records in time sequence, and the date/time fields should continually increase in value. (The TIMEOFF control statement suppresses the date/time field check.)

This message is followed by an MVS abend U111 with a dump.

Chapter 14. System definition file utility program, DFHCSDUP

The CICS system definition utility program, DFHCSDUP, is a component of resource definition online (RDO). DFHCSDUP is an offline utility program that allows you to read from and write to a CICS system definition (CSD) file, either while CICS is running or while it is inactive.

You can use the DFHCSDUP program to:

- ADD a group to the end of a named list in a CSD file
- ALTER attributes of an existing resource definition
- APPEND a group list from one CSD file to a group list in another, or in the same, CSD file
- COPY all of the resource definitions in one group or several generically named groups to another group or several other generically named groups in the same, or in a different, CSD file
- DEFINE a single resource, or a group of resources, on the CSD
- DELETE from the CSD a single resource definition, all of the resource definitions in a group, or all of the group names in a list
- EXTRACT data from the CSD and pass it to a user program for processing
- INITIALIZE a new CSD file, and add to it CICS-supplied resource definitions
- LIST selected resource definitions, groups, and lists
- MIGRATE the contents of a table from a CICS load library to a CSD file
- REMOVE a single group from a list on the CSD file
- SERVICE a CSD file when necessary
- UPGRADE the CICS-supplied resource definitions in a primary CSD file for a new release of CICS
- VERIFY a CSD file by removing internal locks on groups and lists.

You can invoke the DFHCSDUP program in two ways:

1. As a batch program (see page 217)
2. From a user program running either in batch mode or in a TSO environment (see page 219).

Sharing the CSD between CICS/ESA 4.1 and earlier releases

If you want to share the CSD between CICS regions at different release levels, to enable you to share common resource definitions, you must update the CSD from the higher level region – CICS/ESA 4.1.

In CICS/ESA Version 4, some attributes are obsolete, and are removed from the CSD definitions. Using the ALTER command on definitions that specify obsolete attributes does not cause the loss of these attributes in CICS/ESA 4.1, so you can safely update resource definitions from a CICS/ESA 4.1 region. If you are sharing the CSD between a CICS/ESA 4.1 region and a CICS/MVS 2.1.2 or a CICS/OS/VS 1.7 region, you can use the CICS/ESA 4.1 CSD utility, DFHCSDUP,

to update resources that specify obsolete attributes. A compatibility option is added for this purpose, which you must specify on the PARM parameter on the EXEC PGM=DFHCSDUP statement. You indicate the compatibility option by specifying COMPAT or NOCOMPAT. The default is NOCOMPAT, which means that you cannot update obsolete attributes. (See Figure 47 on page 217.)

The *CICS/ESA Migration Guide* discusses these obsolete attributes and their compatibility with earlier releases.

Note: You cannot use the EXTRACT command of the CICS/ESA 4.1 DFHCSDUP utility when the COMPAT option is specified.

Input to the DFHCSDUP program

Input to the DFHCSDUP program (see Figure 46) is from:

- A **primary CSD file**, which must be present, and have a ddname of DFHCSD
- Optionally, a **secondary CSD file**, for which you can specify any ddname
- A CICS table, as specified on the MIGRATE command.

Output from the DFHCSDUP program

The result of running the DFHCSDUP program (see Figure 46) may be an updated primary file, or a print file.

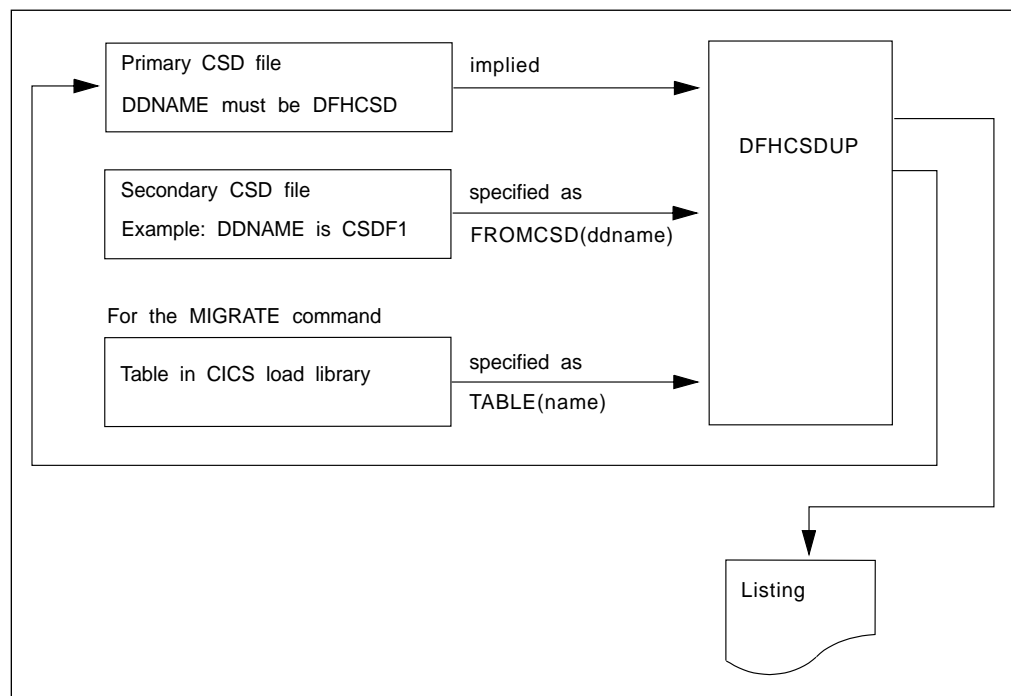


Figure 46. The DFHCSDUP offline utility program

Invoking DFHCSDUP as a batch program

The job in Figure 47 shows you an example of the job control statements you can use to invoke DFHCSDUP as a batch program.

```

//CSDJOB JOB accounting info,name,MSGLEVEL=1
//STEP1 EXEC PGM=DFHCSDUP,REGION=512K,
// PARM='CSD(READWRITE),PAGESIZE(60),NOCOMPAT'
//STEPLIB DD DSN=CICS410.SDFHLOAD,DISP=SHR
//*****
//* If you are running DFHCSDUP with the MIGRATE command,
//* and your CICS load tables are not in CICS410.SDFHLOAD,
//* concatenate your own private library here:
//*****
// DD DSN=CICS410.userlib.tables,DISP=SHR
//DFHCSD DD UNIT=SYSDA,DISP=SHR,DSN=CICS410.DFHCSD
//SECNDCSD DD UNIT=SYSDA,DISP=SHR,DSN=CICS410.SECNDCSD
//indd DD UNIT=SYSDA,DISP=SHR,DSN=extract.input.dataset
//outdd DD UNIT=SYSDA,DISP=SHR,DSN=extract.output.dataset
//* or
//outdd DD SYSOUT=A
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
:
: DFHCSDUP commands
:
/*
//

```

Figure 47. Sample job to run DFHCSDUP

Notes:

1 The EXEC statement should specify a suitable REGION size and a PARM parameter:

- **The REGION size.** A region size of 512KB is generally recommended for the execution of the DFHCSDUP program. However, for the MIGRATE command, the table to be migrated is loaded into main storage, so the region size should be at least 512KB plus the size of the largest table.
- **The PARM parameter.** Use this to specify any of the following options:
 - UPPERCASE** specifies that you want all output from DFHCSDUP to be in uppercase. If you want all output to be in mixed case (the default), do not code this option.
 - CSD({READWRITE|READONLY})** specifies whether you want read/write or read-only access to the CSD from this batch job. The default value is READWRITE.
 - PAGESIZE(nnnn)** specifies the number of lines per page on output listings. Values for nnnn are 4 through 9999. The default value is 60.
 - NOCOMPAT** or **COMPAT** specifies whether the DFHCSDUP utility program is to run in compatibility mode (that is, whether it can update definitions that are obsolete in CICS/ESA 4.1). The default is NOCOMPAT, which means that you cannot update obsolete attributes. For further information about this option, see “Sharing the CSD between CICS/ESA 4.1 and earlier releases” on page 215.

2 You need a DD statement for a secondary CSD if you specify the FROMCSD parameter on an APPEND, COPY, or SERVICE command. The ddname for this DD statement is the name you specify on the FROMCSD parameter. The secondary CSD must be a different data set from the primary; you must not define primary and secondary DD statements that reference the same data set.

3 If you specify the EXTRACT command, you may need to:

- Concatenate with STEPLIB the libraries that contain your USERPROGRAM programs.
- Include a DD statement for any input data set that is defined in your user program. For example, the CICS-supplied user program, DFH\$CRFA, needs a DD statement with a ddname of CRFINPT.

The input file specified by CRFINPT is needed by the user programs DFH\$CRFx (where x=A for Assembler or x=P for PL/I) and DFH0CRFC (for COBOL) to supply the list of resource types or attributes for which you want a cross reference listing. You can specify (in uppercase) any resource type known to CEDA, one resource type per line (starting in column 1). For example, your CRFINPT file may contain the following resource types (one per line) to be cross referenced:

```
PROGRAM
TRANSACTION
TYPETERM
XTPNAME
DSNAME
```

For programming information about the use of the CRFINPT file by the programs DFH\$CRFx or DFH0CRFC (for COBOL), see the *CICS/ESA Customization Guide*.

4 If you specify the EXTRACT command, you need to include the DD statements for any data sets that receive output from your extract program. The ddname is whatever ddname you define in the user program. The CICS-supplied sample programs need DD statements for the following ddnames:

Table 16. DD statements for the CICS-supplied sample programs

program name	ddname	example DD statement
DFH\$CRFx or DFH0CRFC (VS COBOL II)	CRFOUT	//CRFOUT DD SYSOUT=A
DFH\$FORx or DFH0FORC (VS COBOL II)	FOROUT	//FOROUT DD SYSOUT=output.dataset
DFH0CBDC	CBDOUT SYSABOUT	//CBDOUT DD SYSOUT=A //SYSABOUT DD SYSOUT=A

5 The output data sets in these examples are opened and closed for each EXTRACT command specified in SYSIN. If you are writing the output to a sequential disk data set, specify DISP=MOD to ensure that data is not overwritten by successive EXTRACT commands. Alternatively, provided you do not specify SYSOUT on the DD statement, you can change the OPEN statement in the program (for example, in the VS COBOL II versions, to OPEN EXTEND). For programming information about the CICS-supplied user programs, see the *CICS/ESA Customization Guide*.

6 Syntax

You can code commands and keywords using abbreviations and mixed case, as given in the syntax box in the description of each command. If you enter an ambiguous command or keyword, the DFHCSDUP program issues a message indicating the ambiguity.

You can specify keyword values longer than one line, if you use the continuation character (an asterisk) at the end of a line (in column 72). Subsequent lines start in column 1. For example, you can use this facility to specify XTPNAME values of up to 128 hexadecimal characters.

Invoking the DFHCSDUP program from a user program

Invoking the DFHCSDUP program from a user program enables you to create a flexible interface to the utility. By specifying the appropriate entry parameters, your program can cause the DFHCSDUP program to pass control to an exit routine at any of five exit points. The exits can be used, for example, to pass commands to the DFHCSDUP program, or to respond to messages produced by its processing.

You can run your user program:

- In batch mode
- Under TSO.

Notes:

1. In a TSO environment, it is normally possible for the terminal user to interrupt processing at any time by means of an ATTENTION interrupt. In order to protect the integrity of the CSD file, the DFHCSDUP program does not respond to such an interrupt until after it has completed the processing associated with the current command. It then writes message number DFH5618 to the put-message exit, where this is available, and also to the default output file:

```
AN ATTENTION INTERRUPT HAS BEEN REQUESTED
DURING DFHCSDUP PROCESSING
```

Your put-message exit routine can terminate the DFHCSDUP program, if desired. (You **must** supply a put-message routine if you want your operators to regain control after an ATTENTION interrupt.)

2. Suitably authorized TSO users can use the CEDA INSTALL transaction to install resources that have previously been defined with the DFHCSDUP program.

The CICS-supplied sample program, DFH\$CUS1, illustrates how the DFHCSDUP program can be invoked from a user program. It is written as a command processor (CP) for execution under the TSO/E operating system.

The following sections outline the entry parameters of the DFHCSDUP program and the responsibilities of the user program. For programming information about invoking the DFHCSDUP program from a user program, see the *CICS/ESA Customization Guide*.

Entry parameters for the DFHCSDUP program

When invoking the DFHCSDUP program, your program passes a list of up to five parameters, as described below:

OPTIONS

A list of character strings, separated by commas. (The information passed here is that which would otherwise be passed on the PARM keyword of the EXEC statement of JCL.)

Note: A maximum of three options may be specified:

UPPERCASE specifies that you want all output from DFHCSDUP to be in uppercase. If you want all output to be in mixed case (the default), do not code this option.

CSD({READWRITE|READONLY}) specifies whether you require read/write or read-only access to the CSD. The default value is READWRITE.

PAGESIZE(nnnn) specifies the number of lines per page on output listings. Valid values for nnnn are 4 through 9999. The default value is 60.

NOCOMPAT|COMPAT specifies whether the DFHCSDUP utility program is to run in compatibility mode (that is, whether it can update definitions that are obsolete in CICS/ESA 4.1). The default is NOCOMPAT, which means that you cannot update obsolete attributes. For further information about this option, see “Sharing the CSD between CICS/ESA 4.1 and earlier releases” on page 215.

DDNAMES

A list of ddnames that, if specified, are substituted for those normally used by the DFHCSDUP program.

HDING

The starting page number of any listing produced by the DFHCSDUP program. You can use this parameter to ensure that subsequent invocations produce logically numbered listings. If this parameter is not specified, the starting page number is set to 1.

The page number, if supplied, must be four numeric EBCDIC characters.

DCBs

The addresses of a set of data control blocks for use internally by the DFHCSDUP program. Any DCBs (or ACBs) that you specify are used internally, instead of those normally used by the DFHCSDUP program.

Note that if you specify both replacement DDNAMES and replacement DCBs, the alternative DCBs are used, but the alternative DDNAMES are disregarded.

EXITS

The addresses of a set of user exit routines to be invoked during processing of the DFHCSDUP program.

Responsibilities of the user program

Before invoking the DFHCSDUP program, your calling program must ensure that:

- AMODE(24) and RMODE(24) are in force
- S/370 register conventions are obeyed
- If the EXITS parameter is passed, any programming environment needed by the exit routines has been initialized
- Any ACBs or DCBs passed for use by the DFHCSDUP program are OPEN.

Commands for the DFHCSDUP program

This section describes the commands available with the DFHCSDUP utility program. Commands can be abbreviated, but the minimum abbreviation allowed differs from some of the CEDA command abbreviations.

Rules for the syntax and preparation of commands

Enter the commands in columns 1 through 71 of 80-character input records. You can specify keyword values longer than one line, if you use the continuation character (an asterisk) at the end of a line (in column 72). Subsequent lines start in column 1. For example, you can use this facility to specify XTPNAME values of up to 128 hexadecimal characters.

The command keywords can be specified by abbreviations and in mixed case, as shown in the command syntax under each command description. The minimum abbreviation is given in uppercase in the command syntax, with the optional characters given in lower case; for example:

```
ALter Connection(name) Group(groupname)
```

Leading blanks are ignored, and blanks between keywords and operands are permitted.

Comment records are permitted; they must have an asterisk (*) in column 1. Comment material is not permitted on a record that contains a command.

Blank records between commands are ignored.

Follow the conventions for the names of groups and lists when coding the GROUP, LIST, TO, and TYPESGROUP parameters. If you use a generic specification for the GROUP or LIST parameter in the LIST command, you can use the symbols * and + in the same way as for CEDA.

The FROMCSD parameter must contain a valid ddname conforming to the rules for the JCL of the operating system.

An example of a valid sequence of commands is shown in Figure 48 on page 222. Other examples of commands are given in the command descriptions that follow.

```

*                               SET UP INITIAL CSD FILE
INITialize
*
LIst LIst(DFHLIST) Objects
*                               UPGRADE FROM EARLIER RELEASE
UPgrade
*                               MIGRATE MAIN TABLES
MIgrate TABle(DFHFCTF1)
*
MIgrate TABle(DFHTCTT1)
*
LI Group(PPTM1)
LI G(SETM*)
*                               CREATE GROUP PCTZ4
Copy G(PCTM1) To(PCTZ4)
C G(SETMP3) T(PCTZ4) Replace
LI G(P++M+)
*                               CREATE LIST MODLIST
APpend LIst(TESTLIST) TO(MODLIST) FRomcsd(CSDF1)
AP LI(SECLIST) To(MODLIST) FR(CSDF1)
AP LI(DFHLIST) To(MODLIST)
*
LI ALL OBJECTS

```

Figure 48. Sample commands of the DFHCSDUP program

Command processing following internal error detection

If you have provided a put-message-exit routine for the DFHCSDUP program, it is invoked whenever a message is issued. You can use this exit to respond to error messages produced by DFHCSDUP processing, when the DFHCSDUP program is invoked from a user program. The put-message-exit routine is not used if the DFHCSDUP program is running as a batch program. For programming information about the DFHCSDUP exits, see *CICS/ESA Customization Guide*.

The reaction of the DFHCSDUP program to an error (with return code 8 or greater) depends on the nature of the error and on how the DFHCSDUP program is invoked.

If an error is detected while the DFHCSDUP program is running as a batch program, one of the following two reactions occurs:

1. If the error occurs during connection of the CSD, no subsequent commands are completed.
2. If the error occurs elsewhere, no subsequent commands are executed other than LIST commands.

If an error is detected while the DFHCSDUP program is receiving commands from a get-command exit, all subsequent commands are processed if possible.

ADD a group to a list

ADD syntax

```
▶▶—Add—Group(groupname1)—LIst(listname)—▶◀
```

The ADD command adds a group to a list.

Group(*groupname1*)

The name of the group to be added. The name must not already exist in the list. A generic group name is not accepted. If you don't specify a group, the current group name is added.

List(*listname*)

The name of the list to which the group is to be added. If the list does not already exist, a new one is created. If LIST is not specified, the group name is added to the current list if there is one. A generic list name is not accepted.

+
+
+
+

The list name can be up to 8 characters in length. The characters allowed are A-Z, 0-9, @, #, and \$. Lowercase characters are treated as uppercase. Do not use Grouplist names beginning with DFH, because these characters are reserved for use by CICS.

Examples of the ADD command

Create a list LA01, by adding a group to it:

```
ADD GROUP(GA001) LIST(LA01)
```

Add another group to list LA01:

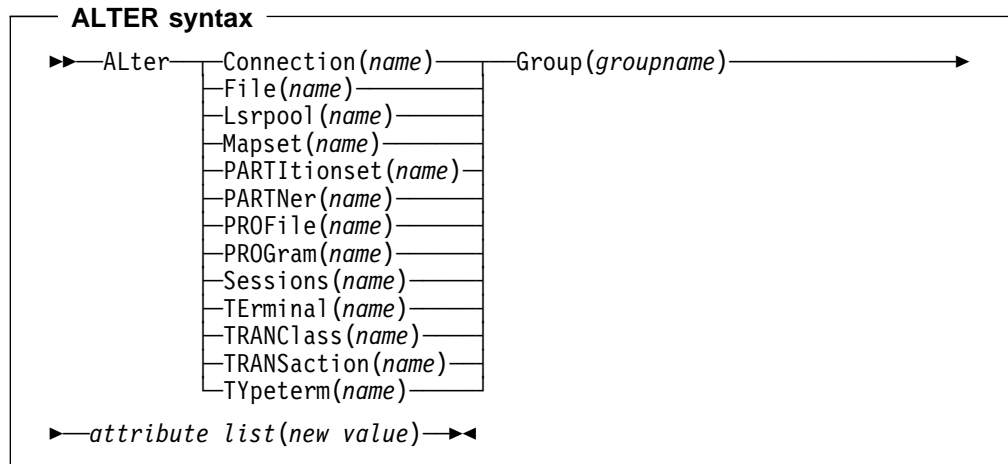
```
ADD GROUP(GA002) LIST(LA01)
```

LA01 now looks like this:

```
GA001
```

```
GA002
```

ALTER a resource definition



The ALTER command changes some or all of the attributes of an **existing** resource definition. A generic resource name or group name is **not** accepted.

For information about the attributes that you can specify on the ALTER command for the various resource types, and for a description of the attributes and default values of each resource type, see the *CICS/ESA Resource Definition Guide*

Resource(name)

The resource whose attributes you want to alter.

Group(groupname)

The name of the group containing the resource to be altered.

Attribute list

The attributes to be altered.

- Use the ALTER command to change some or all of the attributes of an **existing** resource definition.
- Do **not** use ALTER to change the value of the attributes of a TYPETERM definition on which other attributes depend. If you make a mistake with DEVICE, SESSIONTYPE, or TERMMODEL, you should delete the definition, and define a new one with the correct values.

- You can specify null operand values, for example:

```
ALTER FILE(TEST) GROUP(ACT1) DESCRIPTION()
```

If a keyword, for which you have specified a null value, has a default, the default value is used. If a keyword does not have a default value, the definition acts as if the keyword has never been specified. In this example, if you list the resource definition for file TEST, it is shown with DESCRIPTION().

- Changes to resource definitions in the CSD do not take effect until you install the group in which the resource definition resides.

Examples of the DFHCSDUP ALTER command

To make a program resident:

```
ALTER PROGRAM(ERR01) GROUP(GENMODS) RESIDENT(YES) DATALOCATION()
```

APPEND a list to another list

APPEND syntax

```
▶▶—Apend—List(listname1)—To(listname2)—FRomcsd(ddname)—▶◀
```

The APPEND command adds the groups in one list to the end of another list.

List(*listname1*)

Specifies the name of the list that is appended. Do not use a generic list name.

The list being appended can be on the primary CSD, or on another CSD. If you are appending from another CSD, you must identify it by specifying the FROMCSD parameter.

+ The list name can be up to 8 characters in length. The characters allowed are
 + A-Z, 0-9, @, #, and \$. Lowercase characters are treated as uppercase. Do
 + not use Grouplist names beginning with DFH, because these characters are
 + reserved for use by CICS.

To(*listname2*)

Specifies the name of the list to which you want the group names appended. If you are appending from another CSD, you can give this list the same name as the one you are appending from. Do not use a generic list name.

If this target list already exists, the source list is appended to the end of it. If the target list does not exist, it is created. (In effect, you are copying the source list.)

FRomcsd(*ddname*)

Identifies the *ddname* of the secondary CSD from which you are appending *listname1*.

No duplicate group names are allowed in a list. If DFHCSDUP finds any duplicate names during the APPEND operation it ignores them, and they are not appended. The DFHCSDUP output listing contains a warning message if this happens.

Example of the APPEND command

A list called LISTA contains the following groups:

GB001
GB002
GB003

A list called LISTB contains the following groups:

G001
G002
G003

Append LISTB to LISTA, like this:

```
APPEND LIST(LISTB) TO(LISTA)
```

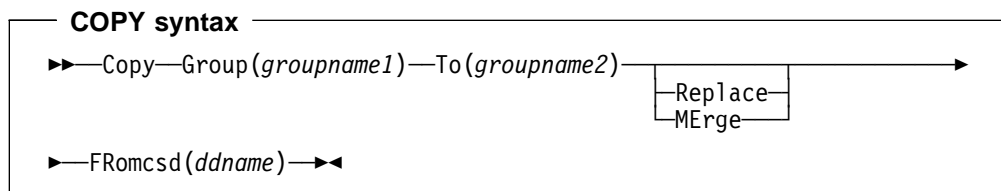
After this, LISTA contains the following groups, in this order:

GB001
GB002
GB003
G001
G002
G003

and LISTB still contains:

G001
G002
G003

COPY a resource definition



The COPY command copies a resource definition, either within the same group or to a different group.

Group(*groupname1*)

Specifies the name of the group to be copied. You can specify a generic name by using an asterisk (*). See “Generic naming in the COPY command” on page 228 for details.

To(*groupname2*)

Specifies the name of the group to which the definitions are copied. If you are copying from another CSD, you can give this group the same name as the one you are copying from. You can specify a generic name by using an asterisk (*). See “Generic naming in the COPY command” on page 228 for details.

Replace

If *groupname2* already exists and duplicate definitions occur, the definitions in *groupname1* replace those in *groupname2*.

MErge

If *groupname2* already exists and duplicate definitions occur, the original definitions in *groupname2* are preserved.

FRomcsd(*ddname*)

Identifies the *ddname* of the secondary CSD from which you are copying *groupname1*.

The COPY command copies all the resource definitions in *groupname1* to *groupname2*. The group to be copied (*groupname1*) can be on the primary CSD, or it can be on the CSD specified by the FROMCSD parameter.

The group is copied to the group named on the TO parameter (*groupname2*) in the primary file. If this group already exists, the definitions from the source group (*groupname1*) are added to those already in the *groupname2* group. If the group specified on the TO parameter does not already exist, a new group of that name is created. However, if duplicate definitions exist in the two groups, the whole copy operation fails unless you specify REPLACE or MERGE to indicate how duplicates should be handled.

Generic naming in the COPY command

The COPY command accepts generic group names, both on the GROUP keyword and on the TO keyword, subject to the following rules:

- The only generic character permitted on the COPY command is the asterisk (*) symbol.
- The prefix length of *groupname1* must be equal to or greater than the prefix length of *groupname2*. Thus COPY GROUP(DFHCOMP*) TO(USRCMP*) is valid, but COPY GROUP(DFHCO*) TO(USRCOMP*) is not.

You can use the asterisk (*) symbol to copy from generically-named groups to other generically-named groups or from generically-named groups to a specific group, as shown in Figure 49.

Note: There is no AS parameter as in the CEDA version of the COPY command.

The DFHCSDUP output listing tells you which definitions were copied, and what happened if duplicates were found.

Examples of the COPY command

The following example copies a group named GA001 to a group named GA002, which already exists, replacing any duplicate resource definitions with those in group GA001.

```
COPY GROUP(GA001) TO(GA002) REPLACE
```

The following example copies group GA003 to group GA004, but if any duplicate definitions occur, preserves the group GA004 definitions.

```
COPY GROUP(GA003) TO(GA004) MERGE
```

The following example copies all the CICS-supplied groups to user-named groups with a prefix of USR, with the result that DFHOPER becomes USROPER, DFHSTAND becomes USRSTAND, and so on.

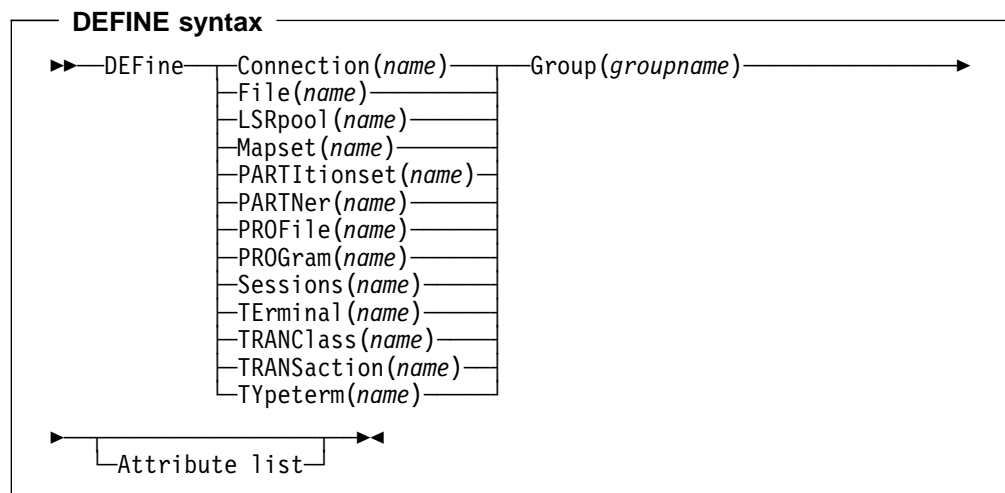
```
COPY GROUP(DFH*) TO(USR*)
```

The following example copies every group starting with ABCD to the group called NEWGROUP:

```
COPY GROUP(ABCD*) TO(NEWGROUP)
```

Figure 49. Examples of the COPY command

DEFINE a resource definition



The DEFINE command creates new resource definitions.

Resource(name)

The name of the resource you want to define. Do not use a generic resource name. The resource keyword must always be the first operand of the DEFINE command.

Group(groupname)

The name of the group containing the resource definition to be altered. Do not use a generic group name. If you specify the name of a group which does not already exist, the group is created.

Attribute list

The attribute list depends on the resource type being defined; some resources have attributes that must be included in the definition. For a description of the attributes and default values of each resource type, Attributes that you do not specify are given default values.

You can use the same name for more than one resource definition in a group, if the definitions are for different resource types. For example:

```

DEFINE PROGRAM(N28A) GROUP(N28APPL)
DEFINE TRANSACTION(N28A) GROUP(N28APPL)

DEFINE TERMINAL(USER) GROUP(USERDEF)
DEFINE PROGRAM(USER) GROUP(USERDEF)
  
```

Examples of the DEFINE command

This example defines two consoles to CICS.
(You do not need continuation symbols if a definition spans several lines).

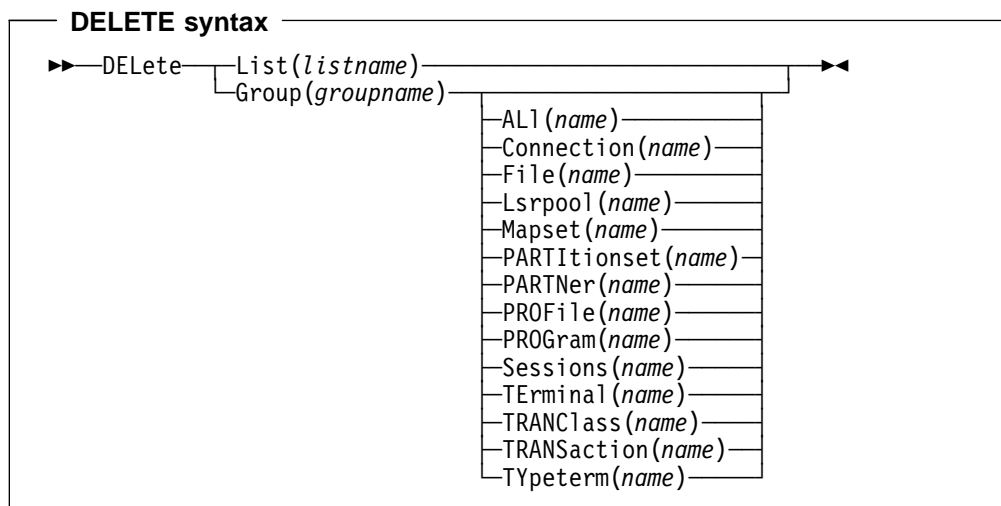
```
DEFINE TERMINAL(CON0)    GROUP(CONTERMS)
      CONSOLE(00)       TYPETERM(DFHCONS)
      DESCRIPTION(MVS CONSOLE ID 00, FOR ISSUING JCL COMMANDS)

DEFINE TERMINAL(CON1)    GROUP(CONTERMS)
      CONSOLE(01)       TYPETERM(DFHCONS)
      DESCRIPTION(MVS CONSOLE ID 01, MVS MASTER CONSOLE)
```

The INITIALIZE command generates a TYPETERM definition, but not a TERMINAL definition, for a console. You must have at least one console defined in order to issue MVS MODIFY commands to CICS. Console id 00 is used to issue commands using MVS job control language, and by authorized programs that use that MGCR macro to issue MVS commands.

Figure 50. Examples of the DEFINE command

DELETE a resource definition, group, or list



The DELETE command deletes a single resource definition in a group, all the resource definitions in a group, or all the group names in a group list.

List(listname)

Specifies the name of the list to be deleted. Do not use a generic list name.

Group(groupname)

If this is specified alone, it indicates the name of the group to be deleted. If a resource is also specified, it indicates the group to which the resource belongs. Do not use a generic group name.

Resource(name)

The name of the resource to be deleted. Do not use a generic resource name.

This parameter can be used only with the GROUP keyword.

Deleting a resource definition is different from removing a group from a list (see “REMOVE a group from a list” on page 241). A deleted resource definition really does disappear from the CSD.

You can not delete the definitions of groups and lists supplied by IBM.

If you delete a list, the definitions of the resources within the groups contained in the list are not deleted. To do this, you must also delete each group individually.

DELETE

Example of the DFHCSDUP DELETE command

A list in the primary CSD called LISTA contains the following groups:

```
GB001  
GB002
```

Group GB001 contains the following resource definitions:

```
TERMINAL(CON0)  
TERMINAL(CON1)  
TERMINAL(TEST)
```

The following command deletes the resource definition for the terminal TEST from group GB001:

```
DELETE TERMINAL(TEST) GROUP(GB001)
```

The following command deletes all the resource definitions in group GB002:

```
DELETE GROUP(GB002)
```

This leaves only group GB001 in the group list LISTA.

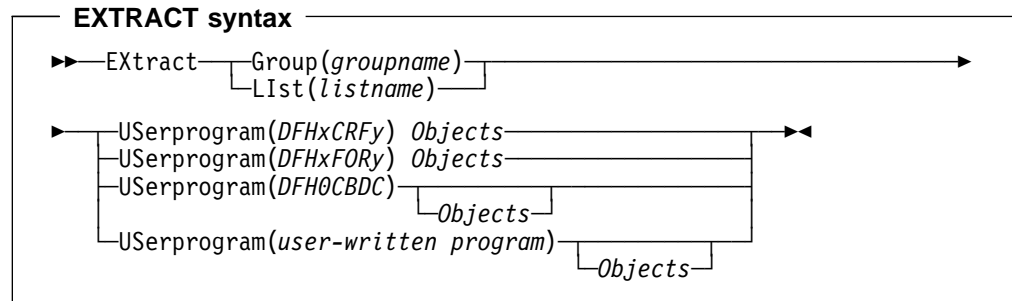
The following command deletes all group names in the group list LISTA:

```
DELETE LIST(LISTA)
```

Note: The resource definitions in the groups in LISTA are not deleted.

Figure 51. Examples of the DELETE command

EXTRACT resource definition data



You can use the EXTRACT command to extract resource definition data from the CSD, either from a list or from a group, and invoke a user program to process the extracted data. You specify the user program on the USERPROGRAM parameter.

Note: For programming information about coding user programs for the EXTRACT command, see the *CICS/ESA Customization Guide*.

Group(groupname)

Selects only those resource definitions within the named group. You can specify a generic group name.

LIst(listname)

Selects only those resource definitions within the groups contained in the named list. You can use a generic list name only if you are not using the OBJECTS option.

Objects

Returns the detail of each resource definition. You can extract resource definition data at two levels of detail:

- Without the OBJECTS option, the command extracts either the names of all the groups within a specified list, or the names of all the resource definitions within a specified group.
- With the OBJECTS option, all the resource definition attributes are also extracted.

You must specify OBJECTS for the CICS-supplied sample user programs DFHxCRFy and DFHxFORy. It is optional for DFH0CBDC and user-written user programs.

USerprogram(user-written program)

Is the name of the user-written program that is to process the data retrieved by the EXTRACT command. You must supply a USERPROGRAM value.

CICS supplies three types of sample user program: DFHxCRFy, DFHxFORy, and DFH0CBDC. The letter x in the program name is \$ for assembler or PL/I and 0 for COBOL. The letter y in the program name denotes the programming language, where y=A is the assembler version, y=C is the COBOL version, and y=P is the PL/I version. Note that DFH0CBDC is supplied as VS COBOL II only. All other user programs are available in source form, in CICS41.SDFHSAMP, and the assembler versions are also available in pregenerated form in CICS41.SDFHLOAD.

EXTRACT

Example of the EXTRACT command

The following command uses the CICS-supplied user program, DFH0CBDC, to extract the resource definitions in group DFHTYPE and create the DEFINE commands needed to create them. It stores these commands in the file specified by the CBDOUT DD statement.

```
EXTRACT GROUP(DFHTYPE) USERPROGRAM(DFH0CBDC) OBJECTS
```

Figure 52. Example of the EXTRACT command

INITIALIZE a new CSD with the required CICS resource definitions

INITIALIZE syntax

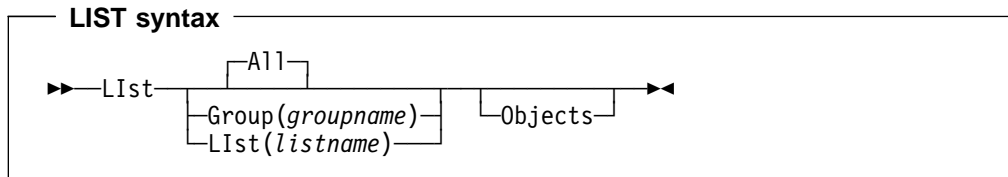
```
▶▶—INITialize—▶▶
```

You can use the INITIALIZE command to prepare a newly defined data set for use as a CSD file. You must initialize your CSD before you can use any of the other DFHCSDUP commands, or the RDO transactions. After you have initialized your CSD, you do not need to execute this function again.

The standard entries for the CICS-supplied resource definitions are created on the CSD. The INITIALIZE command arranges these definitions into groups, and defines these groups in a group list named DFHLIST. This list contains only the CICS-supplied groups that are required by a CICS system.

INITIALIZE also creates a control record at the start of the CSD. This record contains fields identifying the CICS release and the current level of service applied to the CSD. It also has fields containing the date and time of creation of the CSD file, and the date and time the file was last updated. Both these fields appear on the hard copy listing of the CSD produced by the LIST command.

LIST resource definitions



Group(groupname)

Selects only those resource definitions within the named group. You can specify a generic group name.

List(listname)

Selects only those resource definitions within the groups contained in the named list. You can use a generic list name only if you are not using the OBJECTS option (the only command where a generic list name is not acceptable is LIST LIST(listname) OBJECTS).

Objects

This specifies the level of detail required for each resource definition. You can extract resource definition data at two levels of detail:

- Without the OBJECTS option, the command extracts either the names of all the groups within a specified list, or the names of all the resource definitions within a specified group.
- With the OBJECTS option, all the resource definition attributes are also extracted.

The LIST command produces listings of the current status of the CSD. The listings are output to the SYPRINT data set, along with the messages issued by the command processing. The result is to print the contents of all the qualifying groups or lists.

Examples. The listings produced by the various commands are as follows:

- LIST ALL
 - Names of defined lists and groups
 - Summary of lists
 - Summary of groups.

This prints summaries of all the definitions of lists and groups that exist on the CSD.

- LIST ALL OBJECTS
 - Names of defined lists and groups
 - Summary of lists
 - Summary of groups
 - Objects in groups.

This prints summaries of all the definitions of lists and groups that exist on the CSD, together with the properties of the resources in all the groups.

- LIST GROUP(groupname) (group name may be generic)

- Summary of groups.

This summarizes the names of all the resources in one or more groups. They are organized within each group into resource type categories (for example, map sets, programs, and so on).

- LIST GROUP(groupname) OBJECTS (group name may be generic)

- Summary of groups (see above)
- Objects in groups.

This enables you to tabulate the properties of the resources, again organized according to resource type. The creation time for each resource is given, together with all its attributes, as originally set up by using DEFINE and ALTER commands, or by migrating it from a CICS table. The properties of transactions and profiles are arranged in the same subcategories that appear on the CEDA DEFINE screen.

- LIST LIST(listname) (list name may be generic)

- Summary of lists.

The contents of one or more group lists are tabulated. The groups appear in the same sequence as their position in the list. This order is set by the commands ADD and APPEND, which were used in the CEDA transaction to build the list.

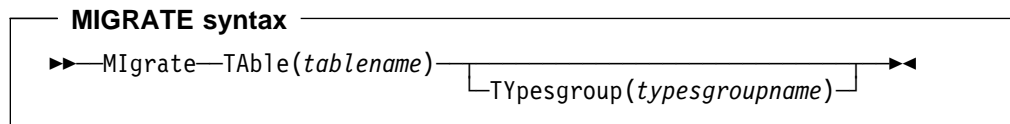
- LIST LIST(listname) OBJECTS (generic list name not allowed)

- Summary of lists (see above)
- Objects of groups in list.

This enables you to tabulate the properties of all the resources to be defined in a CICS system at startup time. These are identified by a single list name in the GRPLIST=listname parameter in the system initialization table. The names of all the groups in the list appear in the summary of lists. Then, for each group contained in the list, the properties of the individual resources in the group are tabulated.

The 'Objects in Groups in Lists' tabulation arranges the groups in the same order as they were added to the group list. This order matters if duplication occurs, when definitions of the same resource may exist in more than one group. If a list of this type is used at system startup time, the resource definitions used when there is duplication are those belonging to the group that is latest in the list.

MIGRATE macro-defined resource definitions from tables to the CSD



The MIGRATE command transfers the contents of an FCT or a TCT from a CICS load library to the CSD.

TAble(*tablename*)

Identifies the name in the load library of the table you want to migrate (that is, DFHFCTxx or DFHTCTxx).

TYpesgroup(*typesgroupname*)

For use with TCT migration only, this is the name of the group to which the TYPETERM definitions are to be migrated.

The contents of a table are transferred as one group, or as a set of several groups, containing definitions. When migrating large tables, make sure you allocate a sufficiently large region for the largest table to be loaded.

+ If you are migrating a table from an earlier release of CICS, you must specify the
 + COMPAT option on the DFHCSDUP PARM of the EXEC statement to prevent
 + migration errors.

- **To transfer an FCT**, the format is:

```
MIgrate TAble(tablename)
```

The result is a set of groups containing file and LSR pool definitions. You can define each group by the macro:

```
DFHFCT TYPE=GROUP, GROUP=xxxxxxxx
```

which you insert in the FCT source instructions before you assemble the FCT for migration. Any file or LSR pool definitions that come before the first such TYPE=GROUP macro are migrated into a group named after the table name: for example, if the table name is DFHFCTxx, the group name is FCTxx.

- **To transfer a TCT**, the format is:

```
MIgrate TAble(tablename) [TYpesgroup(typesgroupname)]
```

where TYpesgroup(*typesgroupname*) specifies the name of the group to contain the TYPETERM definitions obtained from the TCT.

If this parameter is not specified, the TYPETERM definitions are put in the GROUP currently being created, with the TERMINAL definitions.

The result is:

1. A set of groups containing terminal definitions. You can define each group by the macro:

```
DFHTCT TYPE=GROUP, GROUP=xxxxxxxx
```

which you insert in the TCT source instructions before you assemble the TCT for migration. Any terminal definitions that come before the first TYPE=GROUP macro are migrated into a group named after the table name. If the table name is DFHTCTxx, the group name is TCTxx.

2. A group of TYPETERM definitions. These are derived from attributes of TYPE=TERMINAL macros which are often identical for many terminals. They are put into the CSD GROUP named in the TYPESGROUP parameter.

The typeterm attributes of each TYPE=TERMINAL table macro are checked with existing TYPETERM definitions and if they don't match with any of these, a new TYPETERM is added to the CSD.

The existing TYPETERMs checked are:

- TYPETERMs in the GROUP currently being created
- TYPETERMs in the group specified in the TYPESGROUP parameter of the MIGRATE command.

However, the scope of the checking is never extended to include any other TYPETERMs in other groups already on the CSD. (Such groups may have been created using RDO or by a previous MIGRATE command.) For this reason, it is a good idea to use the TYPESGROUP parameter to avoid creating duplicate TYPETERMs in different groups. It is convenient to keep the TYPETERMs in a separate group anyway.

TYPETERMs created on the CSD during the migration are named systematically, in a way related to the TRMTYPE parameter of the original terminal definition. The name consists of a prefix (3–5 characters) with a 3-character suffix. For example, a TYPETERM defining attributes for a 3270 printer is named 3270P001. Variants with the same TRMTYPE are named 3270P002, and so on. The migration process ensures that this name is used as the TYPETERM parameter of every terminal definition that references it.

Note: Migrating your TCT does not cause an error if the destination group already exists. Only definitions that already exist are flagged by an error message; any new or additional definitions are added to the existing group.

Example of the MIGRATE command

An FCT, DFHFCT\$, from CICS/ESA 3.3 contains the following definitions:

```
DFHFCT TYPE=FILE,ACCMETH=VSAM,FILE=DDNAME1,  
        DISP=SHR,JREQ=ALL  
DFHFCT TYPE=FILE,ACCMETH=VSAM,FILE=DDNAME2,  
        DISP=SHR,JREQ=ALL  
DFHFCT TYPE=FILE,ACCMETH=VSAM,FILE=DDNAME3,  
        DISP=SHR,JREQ=ALL
```

To migrate the first and third definitions to the group SAMPLES, and the second definition to the default group, you could add the following DFHFCT macros before the definitions in DFHFCT\$:

```
# DFHFCT TYPE=GROUP,GROUP=SAMPLES before the first definition  
# DFHFCT TYPE=GROUP,GROUP=*DEFAULT before the second definition  
# DFHFCT TYPE=GROUP,GROUP=SAMPLES before the third definition
```

To name the default group STDGROUP, you could use the MIGRATE command:

```
MIGRATE TABLE(DFHFCT$)
```

Figure 53. Examples of the MIGRATE command

REMOVE a group from a list

REMOVE syntax

```
►►—Remove—LIst(listname)—Group(groupname)—►◄
```

The REMOVE command removes a group name from a list. The group, and all its resource definitions, still exists on the CSD.

List(*listname*)

The name of the list from which a group is to be removed. Do not use a generic list name. When the last group is removed from a list, the list no longer exists on the CSD.

Group(*groupname*)

The name of the group to be removed. Do not use a generic group name.

Example of the REMOVE command

A list LL02 contains the following groups:

```
G001 G002 G003 G004
```

To remove group G003:

```
REMOVE GROUP(G003) LIST(LL02)
```

This leaves:

```
G001 G0023 G004
```

Figure 54. Examples of the REMOVE command

SERVICE a CSD

SERVICE syntax

```
►►—Service—FRomcsd(ddname)—LEvel(nnn)—►◄
```

You might occasionally (between CICS releases) have to apply a service routine to carry out preventive or corrective maintenance to your CSD. You do this by loading and running a special service program (DFHCUS1), which is supplied with CICS as a separately loadable module.

You can use the SERVICE command to create a new copy of the CSD, from the existing CSD. All the definitions are preserved, with the corrections (if any) applied.

FRomcsd(*ddname*)

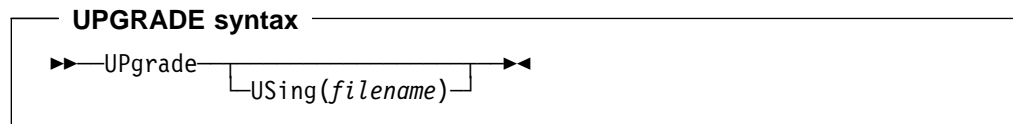
Specifies the *ddname* of the current CSD, which for the purposes of the command is treated as the secondary CSD.

LEvel(*nnn*)

Associated with your CSD is a current service level, initially set to 000 when the file was initialized. Applying the service routine causes the service level to be incremented in steps of one, from a “current level” to a “target level”.

This operand specifies the target service level to which the CSD is to be upgraded, and must be 1 higher than the current level of FROMCSD. Specify it as a 3-character integer; for example, LEVEL(001).

UPGRADE a CSD



You can use the UPGRADE command to change the CICS-supplied resource definitions in a primary CSD. If your existing CSD was initialized at CICS/ESA 3.3 or earlier, you must UPGRADE it using DFHCSDUP to obtain the RDO extensions in CICS/ESA 4.1. Upgrading ensures that the definitions in the DFH-groups are brought up to the level of CICS/ESA 4.1 function.

The upgrade command can also be used to apply any package of IBM-supplied resource definitions to the CSD. For example, the definitions for the CICS sample programs and transactions can be transferred to the CSD with the UPGRADE statement.

USing(filename)

Upgrading a CSD for CICS/ESA 4.1 does not require you to use the USING operand. All IBM-supplied definition from **any** release are deleted and then the CSD is initialized, so you do not need to say which release you came from. However, UPGRADE USING(filename) is used to install IBM features onto CICS. For example, UPGRADE USING(DFHRDJPN) is used to place the double-byte character set feature definitions onto the CSD.

VERIFY a CSD

VERIFY syntax

▶▶—VERIFY—◀◀

The VERIFY command to remove internal locks on groups and lists. Only use it when the CSD is not in use and no backout processing is pending on the CSD; preferably use it only when no CICS systems that may use the CSD are running.

VERIFY acts upon the whole CSD, and is for use in the extreme condition where internal lock records have been left behind. These records are normally removed when a function that changes the CSD has been completed. However, this may not have happened if there was a system failure when the CEDA transaction was running, or if an offline utility failed to finish. The locks may prevent CEDA users from accessing certain groups and lists on the CSD.

Note that VERIFY removes only the internal locks. It does not affect the normal user locks applied by the LOCK command in the CEDA transaction.

Chapter 15. Identify macro-level programs utility program, DFHMSCAN

To convert your CICS applications to command-level, you first have to identify your macro-level programs. To help you do this, CICS provides the DFHMSCAN program to scan a load module library and identify programs that use CICS macros.

Overview

DFHMSCAN scans load modules, looking for instruction sequences that appear to be macro expansions. It locates and, optionally, lists each code sequence that seems to result from a macro instruction. The suspect code sequences may be:

- CICS-supplied DFH macros listed in the *CICS/ESA Application Programmer's Reference (Macro Level)* manual, SC33-0079.
- CICS-supplied macros not listed in the *CICS/ESA Application Programmer's Reference (Macro Level)* manual, but present in MACLIB.
- User-modified CICS macros
- User-written macros
- None of these.

Note that there is no guarantee that a suspect instruction is a CICS-supplied macro, rather than a user or vendor macro or, indeed, none of these. DFHMSCAN's strategy is to list anything that might be a macro, and to cause that part of the program to be examined.

DFHMSCAN identifies CICS DFH macros explicitly where it can. It also reports the use of obsolete EXEC CICS ADDRESS CSA commands.

DFHMSCAN's primary purpose is to give you the information you need to develop a conversion plan, and to quantify the resources you need to achieve it. Based on its reports, you might, for example, decide to convert some of your macro-level programs to command-level, to discard some, and to contact the suppliers of others.

DFHMSCAN does not itself use any CICS macros, commands, or DSECTs. It runs in batch mode and can run concurrently with online CICS systems. It does not alter the contents of the libraries that it scans.

Running DFHMSCAN

To run DFHMSCAN, you need the following JCL:

```
//SCANJOB JOB ACCOUNTING INFO,CLASS=A
//SCAN EXEC PGM=DFHMSCAN,PARM='pppppppp'
//STEPLIB DD DSN=CICS410.SDFHLOAD,DISP=SHR
//INPUT DD DSN=xxxxxxx.LOADLIB,DISP=SHR
//OUTPUT DD SYSOUT=A
//SUMMARY DD SYSOUT=A
//
```

PARM='pppppppp'

The PARM option of the EXEC statement has two possible values, which specify the processing and the report required:

'\$SUMMARY'

The DFHMSCAN program scans every module in the load library, and produces an overall report. This is the default action if PARM is not coded. See “Summary report.”

'NAME1,NAME2,...'

The DFHMSCAN program scans the named modules and produces a detailed report for each one. See “Detailed report” on page 247.

DSN=xxxxxxx.LOADLIB

“xxxxxxx” is the load module library to be scanned. Only one load module library can be specified.

How DFHMSCAN works

DFHMSCAN works by:

1. Loading, one at a time, either all of the programs in the library, or a subset, depending on what value you specify for the PARM option.
2. Scanning each of the specified modules for BALR 14,14, BASR 14,14, and BALR 14,15 instructions.
3. Analyzing the code preceding identified BALR or BASR instructions, to see if it matches sequences produced by CICS macro requests or EXEC CICS commands. DFHMSCAN scans back 20 bytes if it finds a BALR 14,14 or BASR 14,14 instruction, and 40 bytes if it finds a BALR 14,15.
4. Checking each module for EXEC CICS ADDRESS CSA commands.

Using DFHMSCAN

The recommended way to use DFHMSCAN is to:

1. Produce a summary report to identify suspect modules
2. Produce detailed reports to review modules that the summary report flagged as suspect.

Summary report

If you specify PARM='\$SUMMARY', DFHMSCAN summarizes the entire library. The summary report contains:

- A separate analysis of each module in the library:
 - Name
 - Size
 - Language (if determined)
 - Number of CICS macro-level statements
 - Number of CICS command-level statements
 - Number of unrecognized BALR instructions.

If a module seems to contain ADDRESS CSA commands, it is flagged with the message “POSSIBLE ADDRESS CSA”.

- Library-totals of:
 - Modules
 - Macro-level programs
 - Macro-level programs of each type (assembler-language, COBOL, and PL/I)
 - Programs that possibly contain EXEC CICS ADDRESS CSA commands.

Figure 55 is an example of a summary report produced by DFHMSCAN.

DFHMSCAN MODULE	PROGRAM SIZE	SUMMARY LISTING TYPE	ML STMTS	CL STMTS	UR 14,14	UR 14,15	COMMENT
PROGA1	00001B88	ASSEMBLER	2	0	25	1	
PROGP2	00002C3F	PL/I	0	10	10	8	POSSIBLE ADDRESS CSA
PROGC3	00001C54	COBOL	5	0	18	3	
PROGU4	000058CF		0	0	5	15	
TOTAL NO. MODULES SCANNED		NUMBER OF MACRO PROGRAMS	ASSEMBLER MACRO PROGRAMS	COBOL MACRO PROGRAMS	PL/I MACRO PROGRAMS	ADDRESS CSA PROGRAMS	
	4	2	1	1	0	1	

Figure 55. Example of a summary report produced by DFHMSCAN

Detailed report

If you specify PARM='NAME1,NAME2,...', DFHMSCAN scans the named modules only, and produces:

- A detailed report for each named module, that contains:
 - A line for each BALR found, giving:
 - Its offset from the start of the module
 - Its address in storage
 - 20 bytes of the code that precedes it
 - What the code appears to be:

DFHxxx MACRO

A CICS DFHxxx macro, where “xxx” is the two- or three-letter identifier of the macro-type.

DFHxxx call

A specific CICS DFHxxx macro call.

EXEC CICS, EXEC DLI, DLI CALL OR DFHBIF DETECTED

An EXEC CICS or EXEC DLI command, a DLI call, or a DFHBIF macro.

BALR/BASR 14,14 FOUND - NO FURTHER INTERPRETATION

An unidentified instruction, but **not** a CICS-supplied macro. The code may be, for example, a user macro or a user-modified CICS macro that may need to be replaced.

BALR 14,15 FOUND - NO FURTHER INTERPRETATION

An unidentified instruction, but **not** a CICS-supplied macro. The code may be, for example, a user macro or an EXEC CICS command.

- A analysis of the module, in the same form as the analysis of each module in a summary report.
- A summary report for the named modules only.

Limitations of the DFHMSCAN program

The DFHMSCAN program:

- Can scan only one load module for each invocation.
- Does not scan CICS modules and tables in the load library.
- Does not separately identify CHECK macros.
- Cannot identify certain forms of the DFHBIF macro that do not produce a BALR, or that produce code indistinguishable from that generated by EXEC CICS commands.
- Cannot, with certainty, find all EXEC CICS commands for COBOL and PL/I, because the code depends so much on the compiler. Those EXEC CICS commands not flagged by the “EXEC CICS, EXEC DLI, DLI CALL OR DFHBIF DETECTED” message are listed as “BALR 14,15 FOUND - NO FURTHER INTERPRETATION”. The main purpose of the DFHMSCAN program is to find macro-level programs.
- Identifies the language of a module by reference to the language of its last CSECT. This may be confusing if the module is made up of CSECTs written in different languages.
- Prints only 20 bytes of the code preceding a suspect instruction, in its detailed report (even though, for a BALR 14,15 instruction, it has scanned back 40-bytes). This may lead to apparent inconsistencies of interpretation in detailed reports. For example, two similar EXEC CICS commands in a scanned module may produce exactly the same 20-byte output in the report, but be interpreted differently.
- Works by finding code patterns that are similar to those generated by CICS macros. A module can contain such code without having a CICS macro in its source.

Chapter 16. Sample resource manager in-doubt resolution program, DFH\$INDB

You can use the sample program DFH\$INDB in recovery situations to resolve manually any incomplete updates to external resource records

PN88799

The following change was made by APAR PN88799.

+

about which the resource manager, such as DBCTL, DB2 or MQM, is in doubt whether to commit or back out. It produces a list of the in-doubt external resource records, and the action needed to resolve each update.

This manual resolution is needed if CICS is cold-started after the resource manager is abnormally disconnected from CICS. For example, if a CICS region using DBCTL disconnects abnormally from DBCTL (for instance, if CICS or DBCTL abends, or CDBC DISCONNECT IMMEDIATE is issued), there may be some incomplete updates about which DBCTL is in doubt. These in-doubt updates are normally resolved automatically when the CICS region reconnects to DBCTL. However, if for some reason CICS now needs to be cold started, CICS' record of the in-doubt updates is lost and therefore the in-doubt updates are not resolved when CICS reconnects to DBCTL. In this situation the in-doubt updates need to be resolved manually by issuing the appropriate DBCTL operator commands. The DFH\$INDB utility helps this process by producing a list of the DBCTL in-doubt records, and the action needed to resolve each update.

You must run the DFH\$INDB utility using the set of system logs, starting with the system log in use when CICS connected to the resource manager and ending with the system log in use when CICS disconnected abnormally from the resource manager. However, **do not** include system logs that are used for the subsequent cold start of CICS. This means that you may have to run the DFH\$INDB utility **before** you cold start CICS (for example, if your JCT options mean that the system log is reused on the cold start, beginning at the point where the previous CICS run ended). In any case, you should run the DFH\$INDB utility before CICS reconnects to the resource manager. You should use the report produced by the DFH\$INDB utility to resolve the in-doubt updates manually by using operator commands before you reconnect CICS to the resource manager.

Overview of the DFH\$INDB utility

The DFH\$INDB utility is a batch program that reads a series of archived CICS system log data sets and, for a given resource manager, produces a list of in-doubt records and the actions required to resolve them. One line is produced on the report for each in-doubt record, giving the:

- Recovery token
- Name of the CICS transaction being used
- CICS task number
- ID of the terminal on which the transaction was running
- ID of the operator running the transaction
- Time at which the commit processing for the update began

- Action required to resolve the in-doubt.

Inconsistencies between the DFH\$INDB report and actual in-doubt records

Most in-doubt records for a given resource manager appear on the report produced by the DFH\$INDB utility. However, in two cases, a resource manager can be in doubt about a record, but for there to be no corresponding entry in the report produced by the DFH\$INDB utility. These two cases are:

1. The in-doubt record may have been created before the last cold start of CICS. In this case, you must run the DFH\$INDB utility against the archived copies of the logs used prior to the cold start to determine the correct action to resolve the in-doubt record.
2. The resource manager completes its prepare process before CICS writes a 'prepare' record to its log. Therefore, if the resource manager disconnects abnormally from CICS at this point, the resource manager is in doubt, but CICS has no record of it. The correct resolution in this case is always to back-out.

The DFH\$INDB utility can also list in-doubt records that the resource manager does not know about. This can occur if CICS is disconnected abnormally from the resource manager in the time between the resource manager resolving the in-doubt record and CICS writing the 'forget' record to its log. It can also occur if the DFH\$INDB utility is run after resource manager operator commands have been issued to resolve in-doubt records. In these cases, no further action is required.

```
//INDBJOB JOB 'accounting information'CLASS=A,MSGLEVEL=1
//STEP1 EXEC PROC=DFHASMVS,INDEX='CICS410',MOD='DFH$INDB'
//SYSPUNCH DD DSN=&&TEMP,DCB=(RECFM=FB,BLKSIZE=2960),
//          SPACE=(2960,(10,10)),UNIT=SYSDA,DISP=(NEW,PASS)
//SYSIN DD *
:
DFH$INDB source statements
:
/*
//STEP2 EXEC PROC=DFHLNKVS,PARM='LIST,LET,XREF',
//          INDEX='CICS410',INDEX2='your.prefix',NAME='SDFHLOAD'
//SYSPUNCH DD DUMMY
//SYSLIN DD DSN=&&TEMP,DISP=(OLD,DELETE)
//          DD *
//          MODE AMODE(24) RMODE(24)
//          NAME DFH$INDB(R)

/*
//
```

Where
your.prefix is the prefix of your SDFHLOAD library to which
the assembled and link-edited DFH\$INDB utility is written.

Figure 56. Sample job stream to assemble and link-edit the DFH\$INDB utility

Job control statements to assemble and link-edit the DFH\$INDB utility

Before you can run the DFH\$INDB utility, you must assemble and link-edit it into your CICS environment. You can use the CICS-supplied procedure, DFHASMVS, to assemble and link-edit the DFH\$INDB utility. A sample job stream is given in Figure 56.

Job control statements to run the DFH\$INDB utility

Figure 57 gives a sample job stream to run the DFH\$INDB utility. Specify the name of the resource manager, for which you wish to produce the listing of in-doubt records, in the PARM field of the

PN88799

The following change was made by APAR PN88799.

+
+

EXEC statement, for example PARM=DBCTL for DBCTL, PARM=DSNCSQL for DB2 and PARM=MQM for MQSeries. The resource manager name must match the value in the field JCSRNMME in the log record.

The job stream to run the DFH\$INDB utility should include DD statements for the following data sets:

DDname Defines

STEPLIB The load library which contains the DFH\$INDB utility module.

INPUT The archived copies of the CICS system logs that are to be read. Start the sequence of statements for the archived logs with the copy of the log that was in use at the time CICS connected to the resource manager. Include, in chronological order, a statement for a copy of every log data set up to, and including, the log data set that was in use when CICS disconnected abnormally from the resource manager. Specify DCB=RECFM=VB on the DD statement for the first archived log data set.

OUTPUT The data set to which the listing of in-doubt records is written, usually defined to SYSOUT.

```
//RMINDB JOB (accounting information),CLASS=A,  
//      MSGCLASS=A,MSGLEVEL=(1,1)  
//RESOLV EXEC PGM=DFH$INDB,PARM=rmname  
//STEPLIB DD DSN=CICS410.SDFHLOAD,DISP=SHR  
//INPUT   DD DSN=CICS410.logA.copy1,DISP=SHR,DCB=RECFM=VB      1  
//        DD DSN=CICS410.logB.copy1,DISP=SHR                    2  
//        DD DSN=CICS410.logA.copy2,DISP=SHR                    3  
  
:  
//        DD DSN=CICS410.logAorB.copyd,DISP=SHR                 4  
//OUTPUT DD SYSOUT=A  
//SYSABEND DD SYSOUT=A  
//SYSUDUMP DD SYSOUT=A  
//
```

Figure 57. Sample job stream to run the DFH\$INDB utility

Notes: The following sequence of DD statements is used for archived copies of the CICS system logs for this job.

1 CICS410.logA.copy1 – The log in use when CICS connected to the resource manager.

2 CICS410.logB.copy1 – The log used after switching from the previous log (logA.copy1).

3 CICS410.logA.copy2 – The log used after switching from the previous log (logB.copy1).

4 CICS410.logAorB.copyd – The log in use when CICS disconnected abnormally from the resource manager.

You must specify the system logs in the correct order, otherwise the output from the DFH\$INDB utility will be incorrect.

Example of output from the DFH\$INDB utility

Figure 58 shows an example of the output from the DFH\$INDB utility.

```
IN-DOUBT IDENTIFICATION REPORT FOR RESOURCE MANAGER rmmname
LATEST COLD START TOKEN - A17C1A05FEE07100
RECOVERY TOKEN   TRANID TASK_NUM TERMID OPID   FAIL TIME  RESOLUTION ACTION
A17D7A9BBA150401 DLID      31  2B77 0001  12.56.41   COMMIT
A17D6A14FD4D6A01 DLID      35  1E77 0057  11.40.44   BACK OUT
A17D697ED8CB0101 DLUP      32  2D77 X999  11.37.23   COMMIT
END OF REPORT
```

Figure 58. Example report output by the DFH\$INDB utility

Return codes from the DFH\$INDB utility

Code	Meaning
0	Successful completion
4	Resource manager name not specified
8	Supplied resource manager name is more than 8 characters

Source code for the DFH\$INDB utility

For more information about the DFH\$INDB utility, or to create your own version, see the source code in the CICS410.SDFHSAMP library.

Chapter 17. Signon table to RACF migration utility program, DFHSNMIG

The sample utility program DFHSNMIG is provided to transfer operator characteristics from the CICS signon table (SNT) to the RACF database. It reads the SNT and creates a CLIST of appropriate RACF commands (ADDUSER or ALTUSER) for the SNT entries.

Note: The SNT must have been generated at a release of CICS before CICS/ESA 4.1, because CICS/ESA 4.1 does not support generation of SNTs.

The DFHSNMIG program is link-edited as APF-authorized and is installed in the CICS410.SDFHAUTH library. If the DFHSNMIG program is invoked from TSO, add its name to the list of authorized program names in the AUTHPGM NAMES section in the member IKJTSO00 of SYS1.PARMLIB.

Migrating operator characteristics from the CICS SNT to the RACF database

To migrate operator characteristics from your CICS SNT to the RACF database:

1. Edit and submit a job stream to run the DFHSNMIG program. This step creates a CLIST of appropriate RACF commands for the SNT entries to be migrated to the RACF database. You may edit and use the example job stream in “Sample job stream to run the DFHSNMIG program” on page 254.
2. Edit the CLIST to your requirements.

Ensure that the default and non-specified options, such as the password and default RACF group, are suitable for the CLIST. Take care that you do not exceed the maximum number of users in a RACF group; that is, ensure that there is sufficient space remaining in the RACF group to which the SNT entries are to be migrated. (The maximum number of users in a RACF group depends on the length of the user entries.)

3. Edit and submit a job stream to execute the CLIST. This step uses the RACF commands in the CLIST to update the RACF database. You may edit and use the example job stream in “Sample batch job to execute the CLIST” on page 254.

Note: Only a user with the RACF authority SPECIAL can execute the CLIST to update the RACF database.

Sample job stream to run the DFHSNMIG program

```
//SNTMIG JOB 'accounting information',
//          CLASS=A,USER=userid,PASSWORD=password
//SNTMIGR EXEC PGM=DFHSNMIG
//STEPLIB DD DSN=CICS410.SDFHAUTH,DISP=SHR
//DFHSNT DD DSN=CICS330.SDFHAUTH,DISP=SHR
//CLIST DD DSN=CICS410.MIGRATE.CLIST,DISP=(,CATLG),
//          SPACE=(TRK,(10,10),RLSE),UNIT=SYSDA,
//          RECFM=VB,LRECL=256,BLKSIZE=4096
//SYSPRINT DD SYSOUT=A
//*
```

Figure 59. Sample job stream to run the DFHSNMIG program

Figure 59 shows an example job stream to run the DFHSNMIG program. The job stream should include DD statements for the following data sets:

DDname	Defines
STEPLIB	The load library that contains the DFHSNMIG program module.
DFHSNT	The library that contains the DFHSNT module to be migrated. The SNT must have been generated at a release of CICS before CICS/ESA 4.1.
CLIST	The data set to which the CLIST of RACF commands is written. You may change the parameters to suit your requirements, but do not change the parameter RECFM=VB.

Sample batch job to execute the CLIST

```
//RACFMIG JOB 'accounting information',
//          CLASS=A,USER=userid,PASSWORD=password
//DEFINE EXEC PGM=IKJEFT01
//SYSPRINT DD SYSOUT=A
//SYSTSPRT DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//SYSTSIN DD *
EXEC 'CICS410.MIGRATE.CLIST' LIST
/*
//
```

Figure 60. Batch job to execute the CLIST created by the DFHSNMIG program

Chapter 18. Stagger end-of-day time sample utility program, DFH\$STED

Statistics intervals can occur simultaneously across many CICS regions in the same MVS image, and this can degrade performance. To prevent this, you should vary the statistics interval occurrence time for each CICS region. You can use the sample utility program, DFH\$STED, to vary automatically the end-of-day time of each CICS region it is installed in, and therefore vary the interval occurrence time of these CICS regions.

The source code for DFH\$STED is supplied in the *hlq.SDFHSAMP* samples library,
and the pregenerated version is supplied in *hlq.SDFHLOAD*. It uses standard
EXEC CICS calls to set the times and frequencies to produce SMF statistics. The
program source contains extensive comments that explain how the program
functions, and also includes the documented variables. You can use the sample
program as is from SDFHLOAD, or:

- # • Make the appropriate changes for your environment
- # • Assemble the program into a library that is before SDFHLOAD in the DFHRPL
concatenation
- # • Include the CSD group definition for DFH\$STAT into your startup group list
- # • Add the sample program name to the 2nd phase list of programs in your PLTPI
table.

You should run the DFH\$STED program in the third phase of CICS initialization (that is, during the second phase of PLT processing).

+ You can use the following three parameters to control how the end-of-day time is
+ amended. These parameters are part of the source of DFH\$STED. To change
+ them you will have to modify the source of DFH\$STED, which is located in
+ SDFHSAMP.

EODDRIFT Specifies the end-of-day **drift time**; that is, the maximum allowable drift from the original end-of-day time.

This enables you to stagger the end-of-day time of each of your CICS regions by a pseudo-random amount (based upon the time of day at which the program is executing), up to a user-specified maximum value. Since intervals are calculated using the end-of-day as a base time, the occurrence of intervals are staggered by this pseudo-random drift time. The default is ten minutes.

EODTIME Specifies whether the end-of-day time before amendment by the drift value should take the current value (that is, 00:00:00 if COLD started, or the value at previous CICS shutdown if AUTO or WARM started).

You should set this field to CURRENT if you need the current end-of-day time, or FIXED if you need a new end-of-day time. If you specify FIXED, you should specify the new time on the EODFIXED parameter. The default value of the EODTIME parameter is FIXED.

EODFIXED Specifies the new logical end-of-day time, in the form hhmmss, as a hexadecimal value in the range X'000000' through X'235959'. Specify the EODFIXED parameter only if you also specify the EODTIME=FIXED parameter. When used in conjunction with a finite value of EODDRIFT, the drift value specified by the EODDRIFT parameter is applied to the new end-of-day specified by the EODFIXED parameter. When a zero value is specified for the EODDRIFT parameter, the end-of-day time is as specified for the EODFIXED parameter. The default setting is X'000000' (midnight).

Note: If a CICS region is brought up with START=AUTO on several subsequent occasions, and you specify EODTIME=CURRENT, the end-of-day time is never reset, and the drift accumulates.

Example: You could specify the following values for the parameters of the DFH\$STED program if:

- All your CICS regions collect and write their statistics at hourly intervals
- You want to see statistics for all the CICS regions over the same period, but without performance degradation.

```
EODDRIFT=5           (5 minutes maximum drift time)
EODTIME=FIXED        (a new end-of-day time)
EODFIXED=X'000000'   (end-of-day time is midnight)
```

This would vary the statistics intervals by a pseudo-random amount, from midnight, up to maximum of five minutes:

```
Region 1 - statistics taken at 12.00.00
Region 2 - statistics taken at 12.04.10
Region 3 - statistics taken at 12.01.45
  ⋮
Region n - statistics taken at 12.00.27
```

Chapter 19. Message editing utility

This chapter

describes the message editing utility, which you can use to change the text or language of CICS messages, and reassemble the message modules for use by your CICS regions.

Requirements

To be able to use the message editing utility, you need the following:

DASD space

The message editing utility needs 2.5MB for the programs and panels, and 9MB for the source English message data set. The utility allocates 4MB for each target language.

ISPF Version 3

The message editing utility requires a minimum ISPF level of Version 3.

Access authority

To use the message editing utility, you need alter authority for the target data sets index, defined in “Defining the utility data set index” on page 258.

Installing the message editing utility

The library data sets and modules needed by the message editing utility are installed as part of the CICS/ESA 4.1 installation process. (The utility data sets are outlined in “Utility data sets.”) However, before you can use the message editing utility, you must define the ISPF index of the utility data sets. This is described in “Defining the utility data set index” on page 258.

Utility data sets

The following partitioned data sets are used by the message editing utility.

Message source data set

This data set, CICS410.SDFHSRCE, contains all English message source (DFHMExxE) files.

Executable files (CLISTs) data set

This data set, CICS410.SDFHCLIB, contains the message editing utility executable CLIST.

Load library

This data set, CICS410.SDFHLLIB, contains the load modules for the message editing utility.

Messages library

This data set, CICS410.SDFHMLIB, contains the modules for messages internal to the message editing utility.

ISPF panel library

This data set, CICS410.SDFHPLIB, contains the panels for the message editing utility.

Internal tables library

This data set, CICS410.SDFHTLIB, contains the utility-generated tables to control the tracking and processing of message data set members.

Input table of CICS language codes

This data set, CICS410.SDFHLANG, contains the table of all valid language codes supported by CICS.

These data sets, except CICS410.SDFHTLIB, are created automatically when you install CICS/ESA 4.1. The CICS410.SDFHTLIB data set, and some control files, are created automatically when you run the message editing utility. The control files are called `target_data set_index.MEUCNTLx`, where:

target_data set_index

is the index for all message editing utility target data sets.

x

is the CICS one-character language code.

Defining the utility data set index

If, when installing CICS, you change the location of the initial dialog module from CICS410.SDFHCLIB(DFHMEUCL), you must edit DFHMEUCL and change the PROC statement. The PROC statement specifies the location identifier of the utility, and is used to find the initial program for the message editing utility. As supplied, the PROC statement is as follows:

```
PROC 0 MEULIB(CICS410)
```

If you want to invoke the message editing utility with a different data set prefix, you can pass the data set name from the command table. Alternatively, you can use the MEULIB(xxxxxxxx.xxxxxxx) parameter on the CLIST command, where xxxxxxxx.xxxxxxx is the prefix that you want to use. For example:

```
TSO EX 'CICS410.SDFHCLIB(DFHMEUCL)' 'MEULIB(mymeu.prefix)'
```

where *mymeu.prefix* is the prefix to be used for the utility data sets. MEULIB need only be specified if the prefix has been changed from the default.

Using the message editing utility

The process for running the message editing utility is generally:

Step	Task	For details, see 42021dh
1	Start the message editing utility	259
2	Specify your message editing utility default values (the first time run)	260
3	Perform actions on message data sets (from main panel) such as: <ul style="list-style-type: none">• Copy message data set members• Select message sets to be edited• Edit selected messages• Assemble and link-edit changed message set members• Generate a message load module	264
4	Add the new message module to STEPLIB. For each CICS region that is to use the new message module, specify the corresponding language code on the NATLANG system initialization parameter.	268
5	(If needed) Apply PTFs	269

These tasks are described in the following sections, as identified.

Restriction on running the message editing utility: For each target data set index only one user can access the same selected language at a time. However, several users can have the same target data set index for different languages.

A TSO user must not use the message editing utility concurrently from two split screen sessions.

1. Starting the message editing utility

You can start the message editing utility by one of the following methods:

- Add the message editing utility as an option to an ISPF menu.
- Add an entry to the ISPCMDs table. For example:

```
MEU 3 SELECT CMD(EXEC 'CICS410.SDFHCLIB(DFHMEUCL)' 'MEULIB(CICS410)')
```
- Execute as a TSO function from the ISPF command line. For example:

```
TSO EX 'CICS410.SDFHCLIB(DFHMEUCL)' 'MEULIB(CICS410)'
```

Note: If you are starting the message editing utility after it has failed, you must first delete the control files from the previous run before the utility can be restarted. The control files are called `target_data set_index.MEUCNTLx`, where:

target_data set_index is the index for all message editing utility target data sets.

x is the CICS one-character language code.

For more information about using ISPF dialog services to start functions (such as the message editing utility), see the *ISPF Dialog Management Guide and Reference*.

2. Specifying default values for the message editing utility

When you start the message editing utility for the first time, the *Set defaults* panel 1 (of 2) is displayed for you to enter your default values for the utility. For example, see Figure 61 on page 261.

PQ00734

The following note was changed by APAR PQ00734.

Note: When you first start the message editing utility, the following message is overlaid on the CICS macro library and

Apar PQ13574

Documentation for Apar PQ13574 added 31/07/98

CICS SDFHAUTH library lines; but after you press the ENTER key, the message is removed.

MEU017 Defaults must be set before the Message Editing Utility can be used.

While entering the defaults you can press the ENTER key to save the values as you progress. When you have entered all the required default values, you can save the values and exit the panel by using End (F3). This either returns you to the "Message Editing Utility" main panel or displays the *Set defaults* panel 2, as shown in Figure 62 on page 262.

When you have defined your default values, any subsequent start of the message editing utility displays the message editing utility *Main* panel first.

PQ00734

The following figure was changed by APAR PQ00734.

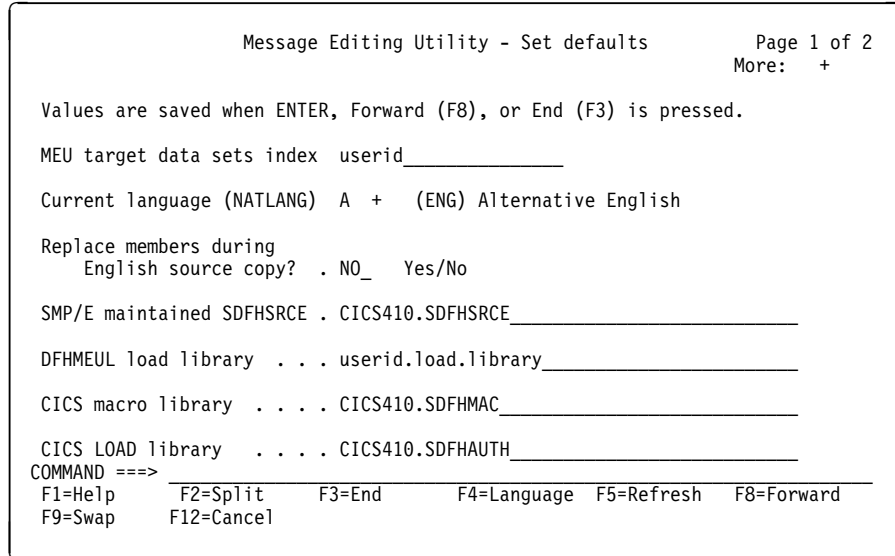


Figure 61. Message editing utility set defaults panel (1 of 2)

Warning

After the *MEU target data sets index* and the *SMP/E maintained SDFHSRCE* parameters have been set they should not be changed. The message editing utility creates a copy of the *SMP/E maintained SDFHSRCE* for its own use, called *MEU target data sets index.SDFHSRCE*. These parameters are also used as a base for the PTF update job. Changing either can result in inconsistencies in the message files.

The + sign beside the current language suffix field indicates that further help is available. Select Language (F4) to view the *Language selection* panel. (See Figure 63 on page 264.) The language suffix that you select is shown in the *Current language (NATLANG)* field.

To refresh the values back to the values last saved, use Refresh (F5).

You are recommended to use one target data set index for all languages. This makes it easier to create message modules for all languages, and to apply any PTF updates for the utility.

To create a message module, the utility needs to find all the translated messages for a language under the same target data set index. Therefore, you must not split the messages modules for a language between data sets with different indexes.

When the utility creates a new message module, it adds the module to the DFHMEUL load library specified on the *Set defaults* panel. For CICS to use this library, it must be APF-authorized, and added to the STEPLIB concatenation of your CICS startup job. (Alternatively, you can copy the new message module to another library in the STEPLIB concatenation.)

PQ00734

The following sentence was changed by APAR PQ00734.

The

Apar PQ13574

Documentation for Apar PQ13574 added 31/07/98

CICS SDFHAUTH library specified on the *Set defaults* panel is used only to find the message module for messages that have not been edited.

```
Message Editing Utility - Set defaults          Page 2 of 2
                                           More: -

Values are saved when ENTER, Backward (F7), or End (F3) is pressed.

JCL output class . . . . . *

Jobcard 1 . . //useridxx JOB (ACCOUNTING INFO),'NAME',_____
           2 . . //_____MSGCLASS=H,_____
           3 . . //_____NOTIFY=userid,_____
           4 . . //_____CLASS=M_____
           5 . . //*_____

Assembler (IEV90 or ASMA90) . . _____

COMMAND ==> _____
           F1=Help      F2=Split      F3=End      F5=Refresh  F7=Backward  F9=Swap
           F12=Cancel
```

Figure 62. Message editing utility set defaults panel (2 of 2). This defines the job statement information added to the JCL to link-edit and generate message load modules, and to apply PTF updates.

Apar PN79349

Documentation for Apar PN79349 added 02/11/98

It also defines which assembler is used in the link-edit JCL.

Apar PN79349

Documentation for Apar PN79349 added 02/11/98

You can specify which assembler the utility uses in the link-edit JCL on the Set defaults panel (2 of 2), shown in figure 62 above. If you intend to use the High Level Assembler, specify ASMA90, otherwise, specify IEV90. These are the only two values accepted.

Selecting languages for message translation: You can select the language to be used in your message set members on the *Language selection* panel. An example of the *Language selection* panel is shown in Figure 63 on page 264. The languages supported by the message editing utility are listed in the *Language selection panel* and for reference in Table 17 on page 263.

Table 17. Languages and codes supported by the message editing utility

NATLANG code	NLS code	Language
A	ENG	Alternative English
Q	ARA	Arabic
1	BEL	Byelorussian
L	BGR	Bulgarian
B	PTB	Brazilian Portuguese
T DBCS	CHT	Traditional Chinese
C DBCS	CHS	Simplified Chinese
2	CSY	Czech
D	DAN	Danish
G	DEU	German
O	ELL	Greek
S	ESP	Spanish
W	FIN	Finnish
F	FRA	French
X	HEB	Hebrew
3	HRV	Croatian
4	HUN	Hungarian
J	ISL	Icelandic
I	ITA	Italian
H DBCS	KOR	Korean
M	MKD	Macedonian
9	NLD	Dutch
N	NOR	Norwegian
5	PLK	Polish
P	PTG	Portuguese
6	ROM	Romanian
R	RUS	Russian
Y	SHC	Serbo-Croatian (Cyrillic)
7	SHL	Serbo-Croatian (Latin)
V	SVE	Swedish
Z	THA	Thai
8	TRK	Turkish
U	UKR	Ukrainian

Notes:

1. **DBCS** denotes Double-Byte Character Set languages.
2. The following language module suffixes are not supported by the message editing utility:
 E - US English master data sets
 K - Japanese data sets, where translation is performed by IBM.
3. A for *alternative English*. Code letter A means "alternative English" to distinguish your edited English message tables from the default US English message tables supplied by CICS. The default US English tables are designated by the language code letter E.
4. You can select only one language a particular message editing utility edit session.
5. The NATLANG code for the selected language is used as the suffix of your edited message data sets to be created from the English language message data sets.

```

                                Language selection   ROW 1 TO 10 OF 33

Use / to select a language, then press ENTER.

  NATLANG  Status  NLS   Language                               DBCS
              code
-   A      Copied  ENG   Alternative English
-   B                                     PTB   Brazilian Portuguese
-   C                                     CHS   Simplified Chinese       DBCS
-   D                                     DAN   Danish
-   F                                     FRA   French
-   G                                     DEU   German
-   H                                     KOR   Korean                       DBCS
-   I                                     ITA   Italian
-   J                                     ISL   Icelandic
-   L                                     BGR   Bulgarian

COMMAND ==> _____ SCROLL ==> PAGE
F1=Help      F2=Split      F3=End      F7=Backward
F8=Forward   F9=Swap        F12=Cancel

```

Figure 63. Message editing utility language selection panel

Languages that have already been set up and used are indicated by the status *Copied*.

To select a language type a / character in the field to the left of the NLS code column and press ENTER.

3. Performing actions on message data sets

You can select message sets to be changed from the *Main* panel of the message editing utility. An example of the *Main* panel is shown in Figure 64 on page 265.

The *Main* panel provides for:

- Copying message data set members
- Selecting message sets to be edited
- Assembling and link-editing changed message set members
- Generating a message load module
- Sorting the list of message set members.

```

Message Editing Utility - Main panel      Row 1 to 47 of 77

Current language: Brazilian Portuguese
MEU index: LARMOUR.PQ17125.CICS510.CREATED.AAA

Type one or more action codes.  Then press Enter.

C Copy  E Edit  L Link-edit

Sort sequence: English name
Last change
Action  English name  New name  Status   Date       Time      Userid    Size
-      DFHMEACE         DFHMEACB Link-edit 1998/07/28 17:23  userid    1205
-      DFHMEAIE         .         .         .         .         .         1827
-      DFHMEAKE         .         .         .         .         .         1216
-      DFHMEAME         .         .         .         .         .         .
-      DFHMEAPE         DFHMEAPA Edited    1998/07/28 17:54  userid    205
-      DFHMEBPE         .         .         .         .         .         .
-      DFHMECCE         .         .         .         .         .         .
-      DFHMECEE         .         .         .         .         .         .
-      DFHMECPE         .         .         .         .         .         .
-      DFHMECRE         .         .         .         .         .         .
-      DFHMEDBE         .         .         .         .         .         .
-      DFHMEDDE         .         .         .         .         .         .
-      DFHMEDEE         .         .         .         .         .         .

COMMAND ==>>> SCROLL ==>>> PAGE
F1=Help      F2=Split    F3=End      F5=Generate F6=Sort     F7=Backward
F8=Forward   F9=Swap     F10=ApplyPTF F11=Defaults F12=Cancel

```

Figure 64. Message editing utility main panel

All the current English message source members are displayed on the *Main* panel. Use Forward (F8) and Backward (F7) to scroll up and down the list.

The status shown for a member is always the last action performed on that member. For example, if the action performed was **Copy** and a previously copied version of the source exists, the status changes to one of the following:

- **Replaced**, if you have set the default parameter *Replace members during English source copy?* to Yes. The English source member is copied to your target source data set.
- **No-Replace**, if the value of the default parameter *Replace members during English source copy?* is No. The English source member is *not* copied to your target source data set. If you wanted to copy the source member, you can change the default and repeat the copy command.

The *Set defaults* panel can be accessed by Defaults (F11). (See Figure 61 on page 261.)

The *PTF update* panel can be accessed by ApplyPTF (F10). (See Figure 67 on page 269.)

The other actions that you can perform from this panel are described in the following sections:

Copying message data set members: To create your own language source member for an English message source member, type C against the member name, then press ENTER. This copies the English source member to your language source data set as a member with the suffix of the current language (as specified on the *Set defaults* panel). For example, with the current language as S (Spanish) and the target data sets index as *target.index*, the member DFHMEACE is copied from the target.index.SDFHSRCE data set to the member DFHMEACS in

the target.index.SDFHSRCS data set. If the target member exists and the replace option on the *Set defaults* panel has been selected, the member in the target data set is replaced and the status changes to Replaced.

Selecting message sets to be edited: To edit messages in a source member, type E against the member name, then press ENTER. This displays the *Message number selection* panel, which lists the messages in the source member and enables you to select messages to be changed. If the selected source member has not previously been copied, requesting the edit action copies the source member before editing it.

Notes:

When editing RP messages, these messages are split between four message sets as follows:

- DFHMEROx for message numbers from 0000 to 0549
- DFHMERPx for message numbers from 0550 to 0999
- DFHMERQx for message numbers from 1000 to 1579
- DFHMERRx for message numbers from 1580 to 9999

When editing ZC messages, these messages are split between four message sets as follows:

- DFHMEZAx for message numbers from 0000 to 2099
- DFHMEZBx for message numbers from 2100 to 3399
- DFHMEZCx for message numbers from 3400 to 5899
- DFHMEZDx for message numbers from 5900 to 9999

When you select a message set to be edited, the utility scans the file for the message numbers that it contains, and displays the *Message number selection* panel for those messages. The message numbers displayed are all the translatable messages from that message set. Any messages that cannot be translated are identified as such by a note added to the message in the *CICS/ESA Messages and Codes*. An example of the *Message number selection* panel is shown in Figure 65 on page 267.


```

Message Editing Utility - Message number selection ROW 1 TO 11 OF 11

Message module: DFHMEACA

Use / to select one or more messages and press ENTER.

  - 2001  - 2002  - 2003  - 2004  - 2005  - 2006
  - 2007  - 2008  - 2009  - 2010  - 2012  - 2014
  - 2015  - 2016  - 2017  - 2018  - 2019  - 2020
  - 2021  - 2022  - 2023  - 2024  - 2025  - 2026
  - 2027  - 2028  - 2029  - 2030  - 2033  - 2034
  / 2035  - 2036  - 2037  - 2038  - 2039  - 2040
  - 2041  - 2042  - 2043  - 2044  - 2047  - 2050
  - 2051  - 2052  - 2053  - 2054  - 2055  - 2056
  - 2057  - 2206  - 2207  - 2208  - 2230  - 2236
  - 2237  - 2238  - 2259  - 2260  - 2261  - 2262
  - 2263  - 2603  - 2605  - 2606  -      -
***** BOTTOM OF DATA *****

COMMAND ==> _____ SCROLL == PAGE_
F1=Help    F2=Split   F3=End     F7=Backward F8=Forward  F9=Swap
F12=Cancel

```

Figure 65. Message editing utility message number selection panel

To select one or more messages to be edited, type a / character in the field to the left of the message number, then press ENTER. When you press ENTER the *Edit message* panel (see Figure 66) is displayed for the messages selected.

Editing selected message sets: You can use the *Edit message* panel to change the text and reply inserts, and the order of inserts, of selected messages. An example of the *Edit message* panel is shown in Figure 66.

```

Message Editing Utility - Edit message

Message number: AC2035

***** TOP OF DATA *****
419000 text " An invalid error code has been passed "
420000 text "to DFHACP. "
421000 text " Transaction "
422000 special_insert tranid
423000 text " is terminated. "
424000 text " Terminal "
425000 special_insert termid
426000 text ". "
***** BOTTOM OF DATA *****

COMMAND ==> _____ SCROLL ==> PAGE_
F1=Help    F2=Split   F3=End     F4=Refresh  F5=Rfind   F6=Rchange
F7=Backward F8=Forward  F9=Swap    F10=Left   F11=Right  F12=Cancel

```

Figure 66. Example of message editing utility edit message panel

For information and rules about editing CICS messages, see “Rules for editing and translating” on page 271.

Assembling and link-editing the changed message data sets: Before you link-edit a source message member, ensure that it is not being used, because this prevents the link-edit job from running.

To link-edit a message source member, type L against the member name, then press ENTER. This submits a job to JES to convert the message source member into assembler language.

Note: Check the link-edit job output to ensure that it completed successfully. If not, examine the error messages generated, correct the message set and re-submit the link-edit job.

Generating a message load module: When you have edited and link-edited all the messages source members that you require, the next step is to create a load module to use with your CICS regions. To do this, use Generate (F5) on the *Main* panel. This assembles the link-edited version of all your message members with the English version of any you have chosen not to translate. Successful completion of this step results in DFHMET1x and DFHMET5x load modules being placed in the data set specified in the DFHMEUL load library field on the *Set defaults* panel. These modules can be used with your CICS job by placing them in the relevant APF-authorized library and specifying the associated language code on the NATLANG system initialization parameter.

Sorting the lists of message set members: To sort the lists of message sets displayed on the *Main* panel use the sort function key (F6). You can sort the message set member list by:

- English name
- New name
- Status
- Date and time.

Each time you press the sort function key the next sort order in the above list is selected.

4. Add the new message load modules to STEPLIB

To enable your CICS region to use the message load module generated by the message editing utility, you must:

- Add the module to a library in the STEPLIB concatenation of your CICS startup JCL.
- Specify the language character suffix of the module on the NATLANG system initialization parameter for your CICS startup job.

Note: CICS always loads the standard English message table by default, regardless of what you specify on the NATLANG system initialization parameter. To ensure your own message tables are selected as the default tables, specify your own language code first on the NATLANG parameter.

Examples: If you modify messages using language code A (for alternative ENGLISH) you should specify NATLANG=A (or NATLANG=(A,E) to ensure your modified message tables are used as the default tables in place of the standard English versions. NATLANG=A is equivalent to NATLANG=(A,E). Do **not** specify (E,A).

If you translate messages using S (for Spanish) and F (for French) and you want French to be the default language with Spanish and English used selectively (by terminal or userid), specify NATLANG=(F,S) or NATLANG=(F,S,E). In this example, if NATLANG is not specified on terminals or userids, French is taken as the default language.

5. Applying PTFs to the message editing utility

This section outlines the process that you use to apply service to the message data sets built via the message editing utility. This is necessary to keep the message files created by the message editing utility in step with the PTF level for your CICS system. It is important that the PTF update process is run whenever you update your CICS PTF level. Failure to do this can result in errors when your running CICS regions issue messages.

To apply PTF updates to your message files:

1. Apply PTFs to the SMP/E-maintained English source. This happens as part of the normal process of applying PTFs to CICS.
2. Select the ApplyPTF (F10) option of the *Main* panel. This displays the *Submit PTF update job* panel is displayed; for example, see Figure 67. To apply a PTF:
 - a. Complete the data set details for the SMP/E maintained SDFHSRCE data set and the update log.
 - b. Press ENTER to validate the input fields.
 - c. Press Submit (F5) to submit the PTF update job.
3. When the update is complete check the output log for messages requiring re-translation.
4. Translate any messages as needed, and re-run the link-edit and generate jobs. This creates a new message module to use with your CICS jobs.

```

      Message Editing Utility - Submit PTF update job      ROW 1 TO 2 OF 2
Change the values below, press ENTER to save or F5 to Submit.

SMP/E maintained SDFHSRCE. CICS410.SDFHSRCE_____
PTF update log . . . . . your.PTFLOG_____
Write log in upper case? . NO_  Yes/No

The PTF updates will be applied to the following data sets.

Data set                                     Language
userid.MEU.SDFHSRCA                          Alternative English
userid.MEU.SDFHSRCM                          Macedonian
***** BOTTOM OF DATA *****

COMMAND ==> _____ SCROLL ==> _____
F1=Help      F2=Split    F3=End      F5=Submit   F7=Backward F8=Forward
F9=Swap      F12=Cancel

```

Figure 67. Message editing utility submit PTF update job panel

When the details have been completed and verified, press the Submit (F5) to instruct the message editing utility to build and submit a TSO CLIST to apply the PTF updates. This CLIST is intended for TSO Background execution only.

Immediately after pressing Submit (F5), the message editing utility terminates. The utility is prevented from restarting while the PTF update job is in progress. If the update process should fail for any reason, two data sets will be left over that will prevent the message editing utility from running. In this situation the following data sets can safely be deleted:

```
userid.MEU.PTFJOB  
userid.MEU.PTFCLIST
```

The submit PTF update job can now be restarted.

Guidelines for PTF update job.

The PTF update job does the following processing:

- New messages are added to all the source data sets.
- Redundant messages are flagged as deleted by placing an * in column 1 of the message definition.
- Changed messages are refreshed with the English message for **all** languages. The old message details are written to the PTF update log.
- All operations are recorded in the PTF update log. (See sample log output in "PTF update log sample output.") Progress messages are output to the console while the CLIST is running.

PTF update log sample output

Figure 68 on page 271 shows a sample of the output for the message editing utility PTF update log.

```

DFHMEUU *** PTF update program started *** yy/mm/dd hh:mm:ss

DFHMEUU PTF COMPARISON STARTED
COMPARING PTF DFHMEACE - ENGLISH DFHMEACE
COMPARING PTF DFHMEAIE - ENGLISH DFHMEAIE
COMPARING PTF DFHMEAKE - ENGLISH DFHMEAKE

DFHMEUU PTF COMPARISON COMPLETED

DFHMEUU -----

DFHMEUU PTF UPDATE STARTED   userid.MEU.SDFHSRCE

DFHMEUU DFHMEAPE UPDATED MESSAGE 0701 - OLD DETAILS FOLLOW
SPECIAL_INSERT APPLID                               07300000
TEXT " AN ABEND (CODE "                               07310000
INS#1 FORMAT CHAR PUBSCHAR "ABCODE"      * 6-CHAR ABEND CODE      07320000
TEXT ") HAS OCCURRED IN EXIT PROGRAM "      07330000
INS#2 FORMAT CHAR PUBSCHAR "PROGNAME"    * 8-CHAR EXIT PROGRAM NAME  07340000
TEXT " AT EXIT POINT "
INS#3                                         * 8-CHAR EXIT POINT NAME
TEXT "."

DFHMEUU DFHMEDEE ADDED   MESSAGE 0118

DFHMEUU DFHMETDE ADDED   MESSAGE 1280

DFHMEUU DFHMETOE DELETED MESSAGE 6024

DFHMEUU PTF UPDATE COMPLETED userid.MEU.SDFHSRCE

DFHMEUU -----

DFHMEUU PTF UPDATE STARTED   userid.MEU.SDFHSRCA

DFHMEUU PTF UPDATE COMPLETED userid.MEU.SDFHSRCA

DFHMEUU *** PTF UPDATE PROGRAM COMPLETED *** yy/mm/dd hh:mm:ss

```

Figure 68. Message editing utility PTF update log sample output

Rules for editing and translating

When editing messages, you **must** observe the following rules.

Message items that must not be altered: You must not alter the following types of message item:

1. ins#n format (CHAR]HEX]DEC]TIME]DATE) pubschar "xxxxx"
2. special_insert xxxxxxxx

These types of inserts must not be changed in any way. However, when editing the message, you can alter the order and position of the inserts within the message, to make the sentence structure more appropriate, but must not change the insert number. The positioning of inserts in the message template determines the location of the inserts in the output message. The suffix #n associates the insert with a variable in CICS code; it does **not** denote its position in the output message. For example:

```

in English
    "text...",ins#1,"text...",ins#2
might be in another language
    ins#2,"text...",ins#1,"text...."

```

```

                                Message Editing Utility - Edit message

Message number: AC2016

***** ***** TOP OF DATA *****
228000 text " Transaction "
229000 special_insert tranid
230000 text " cannot run because program "
231000 ins#1 format CHAR pubschar "program name"
232000 text " is not available."
***** ***** BOTTOM OF DATA *****

COMMAND ==>
F1=Help      F2=Split   F3=End      F4=Refresh  F5=Rfind    F6=Rchange
F7=Backward  F8=Forward  F9=Swap     F10=Left   F11=Right   F12=Cancel

                                SCROLL ==> PAGE_

```

Figure 69. Message editing utility edit message panel showing types of inserts

In the example in Figure 69, line numbers 229000 and 231000 can be moved but must not be altered in anyway.

Message items that can be altered: The message editing utility limits the editing to the message text, to maintain the integrity of the message definition. You can alter the following types of message item:

1. text "text_string"
ins#n format OPT value#n "text_string"

You can translate the text, text_string, which appears between the two double quotes or text delimiters. The "text_string" must not extend beyond column 72 or be continued onto the next line. If more than one line is required for the text, another text "text_string" record must be added. The text may be in upper or mixed case. Double-byte text must be enclosed in shift-out and shift-in delimiters within the text_string.

For optional inserts, OPT value#n, the value#n can spread over several adjacent lines. If you move such an insert, you must move all subsequent value#n lines that are part of the insert. If you do not move all value#n lines for an insert, the message editing utility does not detect this, but CICS will issue an error message if it tries to issued such an incompletely edited message.

An example of this type of message is shown in Figure 70 on page 273. In this example, line numbers 625850 and 625870 must be moved together, and line 625870 must remain below line 625850.

```

Message Editing Utility - Edit message

Message number: SI1502

***** ***** TOP OF DATA *****
625830 text " CICS startup is "
625850 ins#1 format opt value#1 "Cold" value#2 "Warm"
625870 value#3 "Emergency" value#4 "Logterm"
625890 text "."
***** ***** BOTTOM OF DATA *****

COMMAND ==> _____ SCROLL ==> PAGE_
F1=Help      F2=Split   F3=End      F4=Refresh  F5=Rfind    F6=Rchange
F7=Backward  F8=Forward  F9=Swap     F10=Left   F11=Right   F12=Cancel

```

Figure 70. Message editing utility edit message panel, opt insert split over lines

2. reply#n “text_string”

These are a special form of message insert which also serve to define the reply values for a console message requiring an operator reply. They are not be applicable to DBCS languages, because console messages cannot be translated into DBCS languages, unless they are sent to a TDQ destination as well. The positional rules are the same as for other types of inserts. As with the value#n keyword, the text_string within the double quotes following the reply#n keyword may be translated. The text_string **must** be in upper case. An example of this is shown in Figure 71.

```

Message Editing Utility - Edit message

Message number: AP0100

***** ***** TOP OF DATA *****
554200 text " Suffixed module "
554800 ins#1                               format CHAR pubschar "modname"
555400 text " cannot be loaded. ENTER new suffix, "
555600 reply#1 "YES"
555800 text "(unsuffixed), "
556000 reply#2 "NONE"
600000 text "(dummy), or "
640000 reply#3 "CANCEL"
***** ***** BOTTOM OF DATA *****

COMMAND ==> _____ SCROLL ==> PAGE_
F1=Help      F2=Split   F3=End      F4=Refresh  F5=Rfind    F6=Rchange
F7=Backward  F8=Forward  F9=Swap     F10=Left   F11=Right   F12=Cancel

```

Figure 71. Message editing utility edit message panel showing reply#n over several lines

Note for DBCS languages

If a message has a destination of TERMCDBC or CONSOLE then it must not be translated into a DBCS language. If a message has a destination of CONSOLE and TDQ then it can be translated.

Overall message length: The different message destinations have different maximum message lengths. If these are exceeded the message will be truncated. The number of bytes specified for each destination is after the message identifier and default leading inserts have been taken into account, all you need consider is the text you are presented in the *Edit message* panel.

- Console message: Converse messages (that is, those requiring a user response) must not exceed 95 bytes. Other console messages must not exceed 600 bytes.
- Transient data queue messages must not exceed 1200 bytes.

In calculating the overall message length, you must include the lengths of both inserts and text strings. The following is a guide to the lengths of inserts and special_inserts;

- Insert fields (depending on type)

CHAR	n bytes (specified in insert comment)
HEX	up to 14 bytes
DEC	up to 6 bytes
OPT	translatable field of variable length

- Special_inserts

APPLID	9 bytes
SYSID	5 bytes
DATE	9 bytes
TIME	9 bytes
TRANID	5 bytes
TERMID	5 bytes
PROGRAM_NAME	9 bytes
USERID	9 bytes
NETNAME	9 bytes
PRIMARY_ABCODE	5 bytes
SECONDARY_ABCODE	5 bytes

All the special_inserts have a trailing blank which has been taken into account.

Change flags: Some lines have a symbol such as '@PA' at the end of the line. These symbols are IBM internal change flags and can be removed or over typed if needed.

Getting help

From any message editing utility panel, you can press Help (F1) to display help information relevant to that panel.

If you press Contents (F11) from any help panel, the message editing utility *Help contents* panel is displayed.

Message Editing Utility - Help contents

The Message Editing Utility provides a means of editing and translating CICS messages.

Users new to this utility are advised to review the general information topic. For further information refer to CICS/ESA Operations and Utilities Guide.

The following topics may be selected by number:

- 1 - General information
- 2 - Main panel
- 3 - Setting the system defaults
- 4 - Selecting a language suffix
- 5 - Selecting a message number
- 6 - Editing a message
- 7 - Submit PTF update job

COMMAND ==>
F2=Split F9=Swap F12=Cancel

Figure 72. Message editing utility help contents panel

To display help information for a specific topic, type the number for the topic, press ENTER.

Chapter 20. In-doubt window resolution utility, DFH\$IWUP

This chapter

describes the utility program, DFH\$IWUP, which you can use to help determine the resources that have been changed by transactions using ISC or MRO for tasks that are considered to have been in-doubt after a CICS region failure.

Overview of in-doubt windows

If a CICS region fails during a syncpointing process, other CICS regions with which it communicates do not know whether the failed CICS region has backed out or committed its updates. The period from the last syncpoint to the time of failure is known as the **in-doubt window**. For this period, the resource data is potentially inconsistent between the failed CICS region and other CICS regions that also use the same resources.

The danger is that after the failed CICS region has restarted, new transactions can use the inconsistent data; they may read or update the data, and in either case the results of these new transactions can be corrupted.

If a CICS region fails, it issues message DFHZN2101 to identify any units of work, relating to specific tasks, which are considered to be in-doubt. When the CICS region recovers from the failure, it issues further messages to resolve the in-doubt tasks to "In sync" or "Out of sync". (The latter case is reported by message DFHZN2103.) The messages indicate that data integrity may be exposed, but they do not indicate which resources and records are exposed. Also, only CICS resources are reported; not DB2, DBCTL, or other resource types.

The in-doubt window resolution utility program, DFH\$IWUP, helps you to resolve in-doubt resources by listing:

- The user data records, including the resource name and record key if appropriate, that were affected by in-doubt tasks.

This helps you to investigate the messages that report loss of integrity, enabling you to quantify the effect of the failure.

- The resource type, and type of request.

This enables you to differentiate spurious messages from significant ones. (Spurious warning messages of in-doubt problems are issued by CICS regions that have not updated any resources.)

You should run the DFH\$IWUP utility program against copies of the system logs for all CICS regions involved in the units of work that are in-doubt.

Overview of the in-doubt window resolution utility program, DFH\$IWUP

The in-doubt window resolution utility program, DFH\$IWUP, helps you to resolve in-doubt resource updates.

The DFH\$IWUP utility program is an exit program to the CICS journal utility program, DFHJUP. It takes as input a list of unit-of-work IDs as they appear in the DFHZN210x messages generated to the CSMT log.

Note: You can copy the CSMT log and edit it to produce your own list of unit-of-work IDs to be investigated.

The DFH\$IWUP utility program scans the journal, and outputs a report of all journal records for tasks relating to the unit-of-work IDs. The report lists, by task, for each journal record found:

- The resource type and type of request.
This enables you to differentiate spurious messages from significant ones. (Spurious warning messages of in-doubt problems are issued by CICS regions that have not updated any resources.)
- The user data, including resource name and record key (where appropriate).
This helps you to investigate the messages that report loss of integrity, enabling you to quantify the effect of the failure.

To resolve the in-doubt resource update operations, you must interpret the journal data to determine whether or not there has been a loss of integrity.

You can use the DFH\$IWUP utility program to resolve:

- DFHZN2103 messages (for databases out of synchronization) after CICS has reported the result of its recovery logic
- Any units of work whose outcome is unknown after a recovery failure
- Operator action which has prevented CICS from resolving DFHZN2101 messages.

You should run the DFH\$IWUP utility against copies of the system logs for all CICS regions involved in the units of work which are in-doubt.

Note: You cannot use the DFH\$IWUP utility program to resynchronize resources.

Limitations of the DFH\$IWUP utility program

You cannot use the DFH\$IWUP utility program to resynchronize resources.

The DFH\$IWUP utility program does not list unit-of-work IDs for CICS regions connected using LU6.1 protocols. That is, it is unable to resolve any in-doubt resources updated using LU6.1 protocols.

Job control statements to assemble and link-edit the DFH\$IWUP program

Before you can run the DFH\$IWUP program, you must assemble and link-edit it into your CICS environment. You can use the CICS-supplied procedure, DFHASMVS, to assemble and link-edit the DFH\$IWUP program. Figure 73 shows sample job control statements that you can use to assemble and link-edit the DFH\$IWUP utility program.

```
//IWUPASM JOB 'accounting information',CLASS=A,MSGCLASS=A,
//          REGION=2M
//*
//* INVOKE THE DFHASMVS PROCEDURE TO ASSEMBLE DFH$IWUP
//*
//ASM      EXEC DFHASMVS,
//          INDEX=CICS,
//          MOD=DFH$IWUP
//*
//*
//SYSPUNCH DD DSN=&&OBJ,DISP=(,PASS),UNIT=SYSDA,
//          RECFM=FB,BLKSIZE=2960,LRECL=80,
//          SPACE=(CYL,(1,1))
//*
//SYSIN    DD DSN=CICS410.SDFHSAMP(DFH$IWUP),DISP=SHR /* IWUP SOURCE */
//*
//* INVOKE THE DFHLNKVS PROCEDURE TO LINK=EDIT DFH$IWUP
//*
//LKED     EXEC DFHLNKVS,
//          PARM='LIST,LET,XREF',
//          INDEX=CICS,
//          INDEX2='your.prefix', 1
//          NAME=SDFHLOAD
//*
//*
//SYSPUNCH DD DUMMY
//*
//SYSLIN   DD DSN=&&TEMP,DISP=(OLD,DELETE)
//          DD *
//          MODE AMODE(24) RMODE(24)
//          NAME DFH$IWUP(R)
//*
//
```

1 Change your .prefix to the prefix of your SDFHLOAD library to which the assembled and link-edited DFH\$IWUP program is to be written.

Figure 73. Sample JCL to assemble and link-edit the DFH\$IWUP utility program

Job control statements to run the DFH\$IWUP program

To run the DFH\$IWUP utility program, edit and submit the sample JCL shown in Figure 74. Figure 75 on page 281 shows a sample of the report produced by the DFH\$IWUP utility program.

```
//IWUPJOB JOB 'accounting information',CLASS=A,MSGCLASS=A,
//      REGION=2M
//*
//* INVOKE THE JOURNAL UTILITY PROGRAM WITH EXIT DFH$IWUP
//*
//PRINT EXEC PGM=DFHJUP
//STEPLIB DD DISP=SHR,DSN=CICS.SDFHLOAD          /*CICS LIBRARY*/
//      DD DISP=SHR,DSN=your.prefix.SDFHLOAD     /*DFH$IWUP LIBRARY*/
//SYSUT1 DD DISP=SHR,DSN=CICS.J01A              /* JOURNAL IN */
//UOWIDIN DD *                                  /* UOWID IN */
11111111111111111111.X'ttttttttttt'.nnnnn
11111111111111111111.X'ttttttttttt'.nnnnn
/*
//SYSPRINT DD SYSOUT=*,RECFM=FBA,LRECL=133,BLKSIZE=1330
//SYSIN DD *
OPTION PRINT EXITR=DFH$IWUP
END
/*
//
```

Notes: The JCL to run the DFH\$IWUP utility program uses the following DD statements:

DDname Defines

STEPLIB

The CICS load library and, if different, the load library containing the DFH\$IWUP module.

SYSUT1

The journal data set (in use at the time of failure) to be processed.

SYSPRINT

The output data set to which the formatted print records and control messages are to be sent by the DFH\$IWUP utility program.

UOWIDIN

The input list of unit-of-work IDs to be scanned for. Each unit-of-work ID must be entered on a separate line, starting in column 1. Each unit-of-work ID is of the format:

```
11111111111111111111.X'ttttttttttt'.nnnnn
```

where

IIIIIIIIIIIIIIIIII is the LU name

tttttttttt is a hexadecimal token/clock value

nnnnn is the sequence number

Any file of unit-of-work IDs specified on the UOWID DD statement must have an 80-byte record format.

SYSIN

The input control data set for DFHJUP. This file must be in 80-byte record format.

Figure 74. Sample JCL to run the DFH\$IWUP utility program

```

* OPTION PRINT
OPTION PRINT EXITR=DFH$IWUP
*** INVALID UOWID FOUND :   GBIBMIYA.CICSB .X'999999999999'.00001
DFHJC4531 END OF FILE ON INPUT.
DFH$IWFX: CICS JOURNAL PRINT - FORMATTING EXIT
-----
*** UNIT OF WORK ID :   GBIBMIYA.CICSB2.X'B0D53EABCFFC'.00001
-----
*** NO RECORDS FOUND FOR THIS UOWID :
-----
*** UNIT OF WORK ID :   GBIBMIYA.CICSE3.X' AFC6273F5111'.00002
-----
EF59-----SP UNIT OF WORK ID  MODULE-----SYNCH.POINT  RECORD LENGTH-----00055
TASK NUMBER-----00034  TRANSACTION-----ISC2  TERMINAL-----1F77
PHYSICAL START OF TASK-N  LOGICAL START OF TASK--Y  RECORD IN-FLIGHT-----N
EMERGENCY RESTART-----N  DL/I CHECKPOINT-----N  USER PREFIX PRESENT----N
OFFSET FROM 0  DATA  LENGTH-----00025
00001E 000000  180FC7C2 C9C2D4C9 E8C14BC3 C9C3E2C5  F3AFC627 3F511100 02
-----
C311-----FC WRITE ADD  MODULE-----FILE CONTROL  RECORD LENGTH-----00064
TASK NUMBER-----00034  TRANSACTION-----ISC2  TERMINAL-----1F77
PHYSICAL START OF TASK-N  LOGICAL START OF TASK--N  RECORD IN-FLIGHT-----N
EMERGENCY RESTART-----N  DL/I CHECKPOINT-----N  USER PREFIX PRESENT----N
FILE NAME-----FILEA  BASE RBA-----0000000000
OFFSET FROM 0  RECORD ID  LENGTH-----006
DFH$IWFX: CICS JOURNAL PRINT - FORMATTING EXIT
00002A 000000  C9D1D2D3 D4D5
OFFSET FROM 0  DATA  LENGTH-----00016
000030 000000  00100000 40C9D1D2 D3D4D5C9 D1D2D3D4
-----
C311-----FC WRITE ADD  MODULE-----FILE CONTROL  RECORD LENGTH-----00064
TASK NUMBER-----00034  TRANSACTION-----ISC2  TERMINAL-----1F77
PHYSICAL START OF TASK-N  LOGICAL START OF TASK--N  RECORD IN-FLIGHT-----N
EMERGENCY RESTART-----N  DL/I CHECKPOINT-----N  USER PREFIX PRESENT----N
FILE NAME-----FILEA  BASE RBA-----0000000000
OFFSET FROM 0  RECORD ID  LENGTH-----006
00002A 000000  D6D7D8D9 E2E3
OFFSET FROM 0  DATA  LENGTH-----00016
000030 000000  00100000 40D6D7D8 D9E2E3D6 D7D8D9E2
-----
C311-----FC WRITE ADD  MODULE-----FILE CONTROL  RECORD LENGTH-----00064
TASK NUMBER-----00034  TRANSACTION-----ISC2  TERMINAL-----1F77
PHYSICAL START OF TASK-N  LOGICAL START OF TASK--N  RECORD IN-FLIGHT-----N
EMERGENCY RESTART-----N  DL/I CHECKPOINT-----N  USER PREFIX PRESENT----N
FILE NAME-----FILEA  BASE RBA-----0000000000
OFFSET FROM 0  RECORD ID  LENGTH-----006
00002A 000000  E4E5E6E7 E8E9
OFFSET FROM 0  DATA  LENGTH-----00016
000030 000000  00100000 40E4E5E6 E7E8E9E4 E5E6E7E8
-----

```

Figure 75 (Part 1 of 4). Sample output from the DFH\$IWUP program

```

F159-----SP LOGICAL START MODULE-----SYNCH.POINT RECORD LENGTH-----00030
TASK NUMBER-----00034 TRANSACTION-----ISC2 TERMINAL-----1F77
PHYSICAL START OF TASK-N LOGICAL START OF TASK--N RECORD IN-FLIGHT-----N
EMERGENCY RESTART-----N DL/I CHECKPOINT-----N USER PREFIX PRESENT----N
-----
F559-----SP SPR LOG RECORD MODULE-----SYNCH.POINT RECORD LENGTH-----00067
TASK NUMBER-----00034 TRANSACTION-----ISC2 TERMINAL-----1F77
PHYSICAL START OF TASK-N LOGICAL START OF TASK--N RECORD IN-FLIGHT-----N
EMERGENCY RESTART-----N DL/I CHECKPOINT-----N USER PREFIX PRESENT----N
LAST SPR INBOUND---00002 LAST SPR OUTBOUND--00000 OPERATOR ID-----
IN-DOUBT-----N SENT SYS PFX ROOT-----N SUCCESSFUL ABORT-----N
SESSION FAILED-----N WAIT FOR SESS.RECVY----N
OFFSET FROM 0 DATA LENGTH-----00025
00002A 000000 00020000 01000000 C9E2F340 180FC7C2 C9C2D4C9 E8C14BC3 C9
-----
F559-----SP SPR LOG RECORD MODULE-----SYNCH.POINT RECORD LENGTH-----00067
TASK NUMBER-----00034 TRANSACTION-----ISC2 TERMINAL-----1F77
PHYSICAL START OF TASK-N LOGICAL START OF TASK--N RECORD IN-FLIGHT-----N
EMERGENCY RESTART-----N DL/I CHECKPOINT-----N USER PREFIX PRESENT----N
LAST SPR INBOUND---00002 LAST SPR OUTBOUND--00000 OPERATOR ID-----
IN-DOUBT-----N SENT SYS PFX ROOT-----N SUCCESSFUL ABORT-----N
SESSION FAILED-----N WAIT FOR SESS.RECVY----N
OFFSET FROM 0 DATA LENGTH-----00025
00002A 000000 00020000 01000000 C9E2F240 180FC7C2 C9C2D4C9 E8C14BC3 C9
-----
F559-----SP SPR LOG RECORD MODULE-----SYNCH.POINT RECORD LENGTH-----00067
TASK NUMBER-----00034 TRANSACTION-----ISC2 TERMINAL-----1F77
DFH$IWFX: CICS JOURNAL PRINT - FORMATTING EXIT
PHYSICAL START OF TASK-N LOGICAL START OF TASK--N RECORD IN-FLIGHT-----N
EMERGENCY RESTART-----N DL/I CHECKPOINT-----N USER PREFIX PRESENT----N
LAST SPR INBOUND---00001 LAST SPR OUTBOUND--00001 OPERATOR ID-----
IN-DOUBT-----N SENT SYS PFX ROOT-----N SUCCESSFUL ABORT-----N
SESSION FAILED-----N WAIT FOR SESS.RECVY----N
OFFSET FROM 0 DATA LENGTH-----00025
00002A 000000 00010001 01000000 C9E2F140 180FC7C2 C9C2D4C9 E8C14BC3 C9
-----
F559-----SP SPR LOG RECORD MODULE-----SYNCH.POINT RECORD LENGTH-----00067
TASK NUMBER-----00034 TRANSACTION-----ISC2 TERMINAL-----1F77
PHYSICAL START OF TASK-N LOGICAL START OF TASK--N RECORD IN-FLIGHT-----N
EMERGENCY RESTART-----N DL/I CHECKPOINT-----N USER PREFIX PRESENT----N
LAST SPR INBOUND---00002 LAST SPR OUTBOUND--00000 OPERATOR ID-----
IN-DOUBT-----Y SENT SYS PFX ROOT-----N SUCCESSFUL ABORT-----N
SESSION FAILED-----N WAIT FOR SESS.RECVY----N
OFFSET FROM 0 DATA LENGTH-----00025
00002A 000000 00020000 90000000 C9E2F340 180FC7C2 C9C2D4C9 E8C14BC3 C9
-----
F559-----SP SPR LOG RECORD MODULE-----SYNCH.POINT RECORD LENGTH-----00067
TASK NUMBER-----00034 TRANSACTION-----ISC2 TERMINAL-----1F77
PHYSICAL START OF TASK-N LOGICAL START OF TASK--N RECORD IN-FLIGHT-----N
EMERGENCY RESTART-----N DL/I CHECKPOINT-----N USER PREFIX PRESENT----N
LAST SPR INBOUND---00002 LAST SPR OUTBOUND--00000 OPERATOR ID-----
IN-DOUBT-----Y SENT SYS PFX ROOT-----N SUCCESSFUL ABORT-----N
SESSION FAILED-----N WAIT FOR SESS.RECVY----N
OFFSET FROM 0 DATA LENGTH-----00025
00002A 000000 00020000 90000000 C9E2F240 180FC7C2 C9C2D4C9 E8C14BC3 C9
-----

```

Figure 75 (Part 2 of 4). Sample output from the DFH\$IWUP program


```

F559-----SP SPR LOG RECORD  MODULE-----SYNCH.POINT  RECORD LENGTH-----00067
TASK NUMBER-----00034  TRANSACTION-----ISC2  TERMINAL-----1F77
PHYSICAL START OF TASK-N  LOGICAL START OF TASK--N  RECORD IN-FLIGHT-----N
EMERGENCY RESTART-----N  DL/I CHECKPOINT-----N  USER PREFIX PRESENT----N
LAST SPR INBOUND---00000  LAST SPR OUTBOUND--00002  OPERATOR ID-----
IN-DOUBT-----Y  SENT SYS PFX ROOT-----Y  SUCCESSFUL ABORT-----N
SESSION FAILED-----N  WAIT FOR SESS.RECVY---N
OFFSET FROM 0  DATA  LENGTH-----00025
00002A 000000 00000002 D0000000 C9E2F140 180FC7C2 C9C2D4C9 E8C14BC3 C9
-----
F559-----SP SPR LOG RECORD  MODULE-----SYNCH.POINT  RECORD LENGTH-----00067
TASK NUMBER-----00034  TRANSACTION-----ISC2  TERMINAL-----1F77
PHYSICAL START OF TASK-N  LOGICAL START OF TASK--N  RECORD IN-FLIGHT-----N
EMERGENCY RESTART-----N  DL/I CHECKPOINT-----N  USER PREFIX PRESENT----N
LAST SPR INBOUND---00002  LAST SPR OUTBOUND--12624  OPERATOR ID-----
IN-DOUBT-----N  SENT SYS PFX ROOT-----N  SUCCESSFUL ABORT-----N
SESSION FAILED-----N  WAIT FOR SESS.RECVY---N
OFFSET FROM 0  DATA  LENGTH-----00025
00002A 000000 00023150 02000000 C9E2F140 180FC7C2 C9C2D4C9 E8C14BC3 C9
-----
C111-----FC READ UPDATE  MODULE-----FILE CONTROL  RECORD LENGTH-----00064
TASK NUMBER-----00034  TRANSACTION-----ISC2  TERMINAL-----1F77
PHYSICAL START OF TASK-N  LOGICAL START OF TASK--N  RECORD IN-FLIGHT-----N
DFH$IWFX: CICS JOURNAL PRINT - FORMATTING EXIT
EMERGENCY RESTART-----N  DL/I CHECKPOINT-----N  USER PREFIX PRESENT----N
FILE NAME-----FILEA  BASE RBA-----0000000000
OFFSET FROM 0  RECORD ID  LENGTH-----006
00002A 000000 E4E5E6E7 E8E9
OFFSET FROM 0  DATA  LENGTH-----00016
000030 000000 00100000 40E4E5E6 E7E8E9E4 E5E6E7E8
-----
C111-----FC READ UPDATE  MODULE-----FILE CONTROL  RECORD LENGTH-----00064
TASK NUMBER-----00034  TRANSACTION-----ISC2  TERMINAL-----1F77
PHYSICAL START OF TASK-N  LOGICAL START OF TASK--N  RECORD IN-FLIGHT-----N
EMERGENCY RESTART-----N  DL/I CHECKPOINT-----N  USER PREFIX PRESENT----N
FILE NAME-----FILEA  BASE RBA-----0000000000
OFFSET FROM 0  RECORD ID  LENGTH-----006
00002A 000000 D6D7D8D9 E2E3
OFFSET FROM 0  DATA  LENGTH-----00016
000030 000000 00100000 40D6D7D8 D9E2E3D6 D7D8D9E2
-----
C111-----FC READ UPDATE  MODULE-----FILE CONTROL  RECORD LENGTH-----00064
TASK NUMBER-----00034  TRANSACTION-----ISC2  TERMINAL-----1F77
PHYSICAL START OF TASK-N  LOGICAL START OF TASK--N  RECORD IN-FLIGHT-----N
EMERGENCY RESTART-----N  DL/I CHECKPOINT-----N  USER PREFIX PRESENT----N
FILE NAME-----FILEA  BASE RBA-----0000000000
OFFSET FROM 0  RECORD ID  LENGTH-----006
00002A 000000 C9D1D2D3 D4D5
OFFSET FROM 0  DATA  LENGTH-----00016
000030 000000 00100000 40C9D1D2 D3D4D5C9 D1D2D3D4
-----
F159-----SP LOGICAL START  MODULE-----SYNCH.POINT  RECORD LENGTH-----00030
TASK NUMBER-----00034  TRANSACTION-----ISC2  TERMINAL-----1F77
PHYSICAL START OF TASK-N  LOGICAL START OF TASK--N  RECORD IN-FLIGHT-----N
EMERGENCY RESTART-----N  DL/I CHECKPOINT-----N  USER PREFIX PRESENT----N
-----
F459-----SP BAD END OF TASK  MODULE-----SYNCH.POINT  RECORD LENGTH-----00030
TASK NUMBER-----00034  TRANSACTION-----ISC2  TERMINAL-----1F77
PHYSICAL START OF TASK-N  LOGICAL START OF TASK--N  RECORD IN-FLIGHT-----N
EMERGENCY RESTART-----N  DL/I CHECKPOINT-----N  USER PREFIX PRESENT----N
-----

```

Figure 75 (Part 3 of 4). Sample output from the DFH\$IWUP program

```

F559-----SP SPR LOG RECORD  MODULE-----SYNCH.POINT  RECORD LENGTH-----00067
TASK NUMBER-----00034  TRANSACTION-----ISC2  TERMINAL-----1F77
PHYSICAL START OF TASK-N  LOGICAL START OF TASK--N  RECORD IN-FLIGHT-----N
EMERGENCY RESTART-----N  DL/I CHECKPOINT-----N  USER PREFIX PRESENT-----N
LAST SPR INBOUND---00001  LAST SPR OUTBOUND--00000  OPERATOR ID-----
IN-DOUBT-----N  SENT SYS PFX ROOT-----N  SUCCESSFUL ABORT-----Y
SESSION FAILED-----N  WAIT FOR SESS.RECVY---N
OFFSET FROM 0  DATA  LENGTH-----00025
00002A 000000 00010000 20000000 C9E2F340 180FC7C2 C9C2D4C9 E8C14BC3 C9
-----
F559-----SP SPR LOG RECORD  MODULE-----SYNCH.POINT  RECORD LENGTH-----00067
TASK NUMBER-----00034  TRANSACTION-----ISC2  TERMINAL-----1F77
PHYSICAL START OF TASK-N  LOGICAL START OF TASK--N  RECORD IN-FLIGHT-----N
EMERGENCY RESTART-----N  DL/I CHECKPOINT-----N  USER PREFIX PRESENT-----N
DFH$IWFX: CICS JOURNAL PRINT - FORMATTING EXIT
LAST SPR INBOUND---00001  LAST SPR OUTBOUND--00000  OPERATOR ID-----
IN-DOUBT-----N  SENT SYS PFX ROOT-----N  SUCCESSFUL ABORT-----Y
SESSION FAILED-----N  WAIT FOR SESS.RECVY---N
OFFSET FROM 0  DATA  LENGTH-----00025
00002A 000000 00010000 20000000 C9E2F240 180FC7C2 C9C2D4C9 E8C14BC3 C9
-----
*** NO MORE RECORDS TO PRINT FOR THIS UOWID
-----
*** END OF SEARCH. NO MORE RECORDS TO PRINT
-----
DFHJC4532 OPTION COMPLETE.
DFHJC4527 END OF JOB.

```

Figure 75 (Part 4 of 4). Sample output from the DFH\$IWUP program

Error messages produced by the DFH\$IWUP program

The following messages may be issued by the DFH\$IWUP utility program:

ERRMSG02 PROGRAM HAS TERMINATED- NO VALID unit-of-work IDS SUPPLIED.

Explanation: Either no unit-of-work IDS were specified or the unit-of-work IDs supplied were not of the correct format.

System Action: Processing terminates.

User Response: Correct the unit-of-work IDs and resubmit the DFH\$IWUP job.

Destination: SYSPRINT

job statement: UOWIDIN DD statement

ERRMSG04 PROGRAM HAS TERMINATED- OUT OF MEMORY. UNABLE TO BUILD THE LIST OF UOWIDS.

Explanation: While building the list of UOWIDs the DFH\$IWUP utility program ran out of memory.

System Action: Processing terminates.

User Response: Resubmit the DFH\$IWUP job with either a larger REGION size or a smaller list of UOWIDs.

Destination: SYSPRINT

job statement: EXEC PGM=DFHJUP statement or UOWIDIN DD statement

ERRMSG08 OUT OF MEMORY. UNABLE TO SAVE FURTHER RECORDS. PROGRAM WILL TERMINATE AFTER PRINTING ITEMS FOUND SO FAR

Explanation: While searching the journal, the DFH\$IWUP utility program ran out of memory.

System Action: Processing terminates.

User Response: Resubmit the DFH\$IWUP job with either a larger region or a smaller list of UOWIDs.

Destination: SYSPRINT

job statement: EXEC PGM=DFHJUP statement or UOWIDIN DD statement

ERRMSG16 PROGRAM HAS TERMINATED- UNABLE TO OPEN UOWIDIN FILE.

Explanation: An error occurred while opening the UOWIDIN file.

System Action: Processing terminates.

User Response: Check the JES job log for any associated messages from the OPEN macro. Correct and resubmit the DFH\$IWUP job.

Destination: SYSPRINT

job statement: UOWIDIN DD statement

ERRMSG32 PROGRAM HAS TERMINATED- UNABLE TO READ ITEM FROM UOWIDIN FILE.

Explanation: An input/output error occurred while reading the UOWIDIN file.

System Action: Processing terminates.

User Response: Check the JES job log for any associated messages. Correct and resubmit the DFH\$IWUP job.

Destination: SYSPRINT

job statement: UOWIDIN DD statement

Creating your own DFH\$IWUP utility program

You can create your own DFH\$IWUP utility program, for which the source code is supplied in member DFH\$IWUP of the CICS410.SDFHSAMP library.

Apar PQ07674

Documentation for Apar PQ07674 added 27/10/98

Chapter 21. Local catalog storage program, DFHSMUTL

The local catalog storage manager domain subpool record manipulation program,
DFHSMUTL, is an MVS batch program that adds or removes storage manager
domain subpool records to or from the CICS local catalog data set. The records
are used to store tuning information that CICS uses to determine the optimum sizes
of the subpools, and to indicate to CICS which subpools are to have the self-tuning
mechanism enabled.

If the local catalog is re-initialized, DFHSMUTL should be run again to add the
required subpool records to the local catalog.

Input

Control statements are read from SYSIN that specify the subpool records to be
added to the local catalog data set.

The format of the control statements is as follows:

- # 1. All commands must start in column 1.
- # 2. An asterisk in column 1 indicates a comment. Everything else on the line is
ignored.
- # 3. To add a new subpool record to the local catalog, code ADD
SUBPOOL=**name**, where **name** is the name of the subpool the record is added
to (for example, ADD SUBPOOL=ZCTCTUA). There can only be one blank
after ADD, and the rest of the line following the subpool name must be blank.
The subpool name is not checked for validity.
- # 4. To delete a subpool record from the local catalog, code DEL SUBPOOL=**name**,
where **name** is the name of the subpool the record is deleted from (for
example, DEL SUBPOOL=ZCTCTUA). There can only be one blank after DEL
and the rest of the line following the subpool name must be blank.
- # 5. To print a list of subpool records from the local catalog, code LST.

Deleting and adding a subpool record resets the tuning information for that subpool.

Output

Storage manager domain subpool records that are written to the CICS local catalog
data set.

Messages

Messages, including errors, are written to SYSPRINT. The DFHSM0300
DFHSMUTL REPORT shows:

ADD SUBPOOL=xxxxxxx PROCESSED SUCCESSFULLY (ADD
SUBPOOL=xxxxxxx has been processed successfully.)

DEL SUBPOOL=xxxxxxx PROCESSED SUCCESSFULLY (DEL
SUBPOOL=xxxxxxx has been processed successfully.)

```

#          FOUND DFHLCD RECORD SMSUBPOL=xxxxxxx (Subpool record found by
#          the LST command.)
#
#          ERROR OPENING DFHLCD (An error has occurred opening the local catalog
#          data set. The program is terminated.)
#
#          UNRECOGNISED VERB xxx IN INPUT (Only ADD, DEL and LST are allowed.
#          The statement is ignored.)
#
#          UNRECOGNISED OPERAND xxxxxxxx IN INPUT (Only ADD
#          SUBPOOL=xxxxxxx or DEL SUBPOOL=xxxxxxx are allowed. the statement
#          is ignored.)
#
#
#          ERROR PROCESSING 'ADD SUBPOOL=xxxxxxx'.
#          R15 = X'yy'.
#          RPL FEEDBACK CODE = X'zz'.
#          SEE DFSMS/MVS MACRO INSTRUCTIONS FOR DATA SETS
#
#          (A VSAM error has occurred while processing an ADD SUBPOOL=xxxxxxx
#          command. For the meaning of the VSAM codes, refer to DFSMS/MVS Macro
#          Instructions for Data Sets, SC26-4913. The program is terminated.)
#
#
#          ERROR PROCESSING 'DEL SUBPOOL=xxxxxxx'.
#          R15 = X'yy'.
#          RPL FEEDBACK CODE = X'zz'.
#          SEE DFSMS/MVS MACRO INSTRUCTIONS FOR DATA SETS
#
#          (A VSAM error has occurred while processing a DEL SUBPOOL=xxxxxxx
#          command. For the meaning of the VSAM codes, refer to DFSMS/MVS Macro
#          Instructions for Data Sets, SC26-4913. The program is terminated.)
#
#          END OF DFHSMUTL REPORT (Report trailer.)

```

Job control statements to run the DFHSMUTL program

```

#          //SSYLCD JOB (accounting information),
#          // CLASS=A,MSGCLASS=A,MSGLEVEL=(1,1),USER=userid,NOTIFY=userid
#          //*****
#          //*
#          //* Use DFHSMUTL to add or remove storage manager domain
#          //* subpool records to or from the local catalog data set
#          //*
#          //*****
#          //SMUTL EXEC PGM=DFHSMUTL
#          //STEPLIB DD DSN=CICS410.SDFHLOAD,DISP=SHR
#          //SYSPRINT DD SYSOUT=*
#          //SYSUDUMP DD SYSOUT=*
#          //DFHLCD DD DSN=CICS410.applid.DFHLCD,DISP=OLD
#          //SYSIN DD *
#          ADD SUBPOOL=name-of-subpool-to-add
#          DEL SUBPOOL=name-of-subpool-to-remove
#          LST
#          /*
#          //

```

1

2

3

Note:

- # **1** Change CICS410.applid.DFHLCDC to the name of the local catalog.
- # **2** Example, to enable the self-tuning mechanism for the ZCTCTUA subpool, specify ADD SUBPOOL=ZCTCTUA.
- #
- # **3** Example, to disable the self-tuning mechanism for the ZCTCTUA subpool, specify DEL SUBPOOL=ZCTCTUA.
- #

Appendix A. Operating procedures

This appendix describes the procedures that we suggest you follow to start and run an online CICS region.

In the descriptions of the operating procedures, we use various departmental and job titles. You may not have a department or job with each of the titles used, so we also give definitions in "Departmental and job titles" to enable you to identify the department or individual that performs the function in your organization.

Various forms and tables are illustrated to help you operate the master terminal. These are for your guidance; you may need to modify them for your organization. Some of the forms are intended for your database and data communication administration people. Others are for you to complete to document the online session. The forms shown in this chapter are:

- CICS online sessions log sheet
- CICS incident report
- Data set, database and transaction table
- Transaction and program table.

Departmental and job titles

The following are definitions of the departmental and job titles that are used in subsequent descriptions of operating procedures.

Application supervisor

The person responsible for resolving errors in designated applications on production systems.

Database (DB) administration

The department responsible for the design, development, operation, integrity, maintenance, and use of databases.

Data communication (DC) administration

The department responsible for the administration of online interactive systems.

Job scheduler

The person responsible for determining the order in which jobs are run, taking into account priorities and interdependences.

Network controller

The person responsible for the provision and maintenance of the hardware equipment (for example, terminals and lines) through which connections are made between terminals and hosts.

User liaison

The department responsible for communication between data processing functions and end users.

CICS ONLINE SESSIONS LOG SHEET						
Job name		Date	Time	Sys. Operator	MTO name	Run No.
	Start					
CICS tables -	Stop					
Instructions.	Other special instructions					
SIT						
DCT						
JCT	Log tape(s)					
MCT		VOLID if SLT jnl	Physical labels	CICS allocated labels yyddd/mm RUN hhmss		Status/note
				RUN		
				RUN		
				RUN		
				RUN		
Start type		Log tape(s) from previous CICS run in case of emergency restart as displayed on console				
AUTO	COLD					
Program not found						
Transaction disabled						
Terminal unavailable						
File unable to open						
Database stopped						
Time	Comments/incidents					
At startup						
At shutdown						

Figure 76. CICS online sessions log sheet

CICS online sessions log sheet

Use this log sheet to document each online session. You should complete it every time you start up CICS with the information needed to restart the system.

This CICS online sessions log sheet is submitted to operations by DC Administration, or the job scheduler, with some information already filled in. This information typically includes the job name, date, run number, physical labels of available log tapes, and any special instructions for the startup.

DC Administration specifies the start type required (usually START=AUTO) and, if START=AUTO results in an emergency restart, write in the label(s) of any log tape(s) from the previous run. This information comes from the log sheet of the previous run.

When starting up the system, you should do the following:

1. Record the start time and your name.
2. If you are using **standard-labeled tape journals**, a specific tape volume is requested when the journal file is being opened for output. You can record the volume identifier together with the CICS-supplied label information on the log sheet. You do not need a sticker label in this case.

If you are using **unlabeled tape journals**, a scratch tape volume is requested when the journal file is being opened for output. Mount the first tape from the available log tape list; CICS allocates a label for it. Record the allocated label on the log sheet. Do not forget to prepare a sticker with this label for use when the tape is unloaded during or at the end of the session. You may use more than one log tape.

3. If you perform an emergency restart and use tape rather than disk data sets for the system log, you need the system log tapes from the previous session. Start by mounting the last log tape from the previous CICS run.

For standard-labeled tape journals, the operating system asks you to mount a specific tape volume by referring to its volume identifier. If the tape volume has already been mounted, however, this message is not issued.

For unlabeled tape journals, DC Administration should already have entered the label of this tape (taken from the previous log sheet), on the current log sheet. When you mount the tape, CICS displays the first record label on the console for you to verify.

Document other useful information; for example, programs not found in the library, disabled transactions, files that could not be opened, or network problems.

Use the bottom portion of the form, "Comments/incidents," to record any unusual incidents that occur at startup, shutdown, or during the session. Note here incidents such as hardware errors on lines or terminals, or abending transactions. Cross-refer to the CICS incident report where necessary. Some of these incidents appear on the system console and others are reported as CICS messages to a destination determined by the installation. This may be the console, another terminal, or a hard-copy device.

CICS incident report

You should complete this form in duplicate to record any of the following error conditions:

- MVS abend
- CICS abend
- Application abend
- Line/terminal hardware problem
- I/O errors on databases
- I/O errors on journal files
- Response time problems reported by users.

Note the sequence number of the CICS incident report on the CICS online log sheet for that session.

In the middle block under the headings “Hardware” and “Software”, briefly describe the symptoms of the problem. If a dump is available, indicate the type of dump and the time it was taken, and, once it has been listed, the time of printing. If possible, describe the problem more fully in the bottom part of the form, and any action you have taken. This “Action” column may also be used by the recipient.

Keep one copy of the form for follow-up purposes and give the other copy to one of the following:

- Your network control people (if the incident was caused by a line or terminal error)
- The appropriate supervisor (if the incident was caused by an application problem)
- Your DC administration (in all other cases).

Data set/database and transaction table

This table is prepared initially by DC Administration. Your user liaison people should also use it, for example, whenever they ask you to disable transactions that use a particular data set or database.

The table shows you the relationship between the data set name and the IMS database names or the CICS file names, the volume serial numbers of the packs containing those data sets, the access method type, and the applications that use them. It also shows the relationship of all transactions to each data set or database.

The table also indicates whether each VSAM data set is eligible for backup-while-open (BWO). For background information about backup-while-open, see the *CICS/ESA System Definition Guide* and the *CICS/ESA Recovery and Restart Guide*.

From this table you can tell which transactions to disable before taking a data set or database offline. The date, time, and name at the lower left corner are used when you disable a related transaction at the request of your user liaison people.

Transaction and program table

This table is prepared by DC Administration.

It shows you the relation between transaction codes and programs, the databases used, and whether a transaction accesses the data sets or databases for update or read-only purposes. It also shows the task class, the programs invoked by a transaction, the application, and the status of the program, for example, "Enabled" or "Disabled."

The date, time, and name at the lower left hand corner is used when you disable a transaction at the request of your user liaison people.

Error procedures

Some user problems and system failures that require you to establish error procedures are illustrated in the following flowcharts:

Flowchart

A-1 (see page 300)

A-2 (see page 302)

Procedure for

Machine check, power failure, MVS loop, orabend

Action on user problem.

Chart A-1: Machine check, power failure, MVS loop, or abend

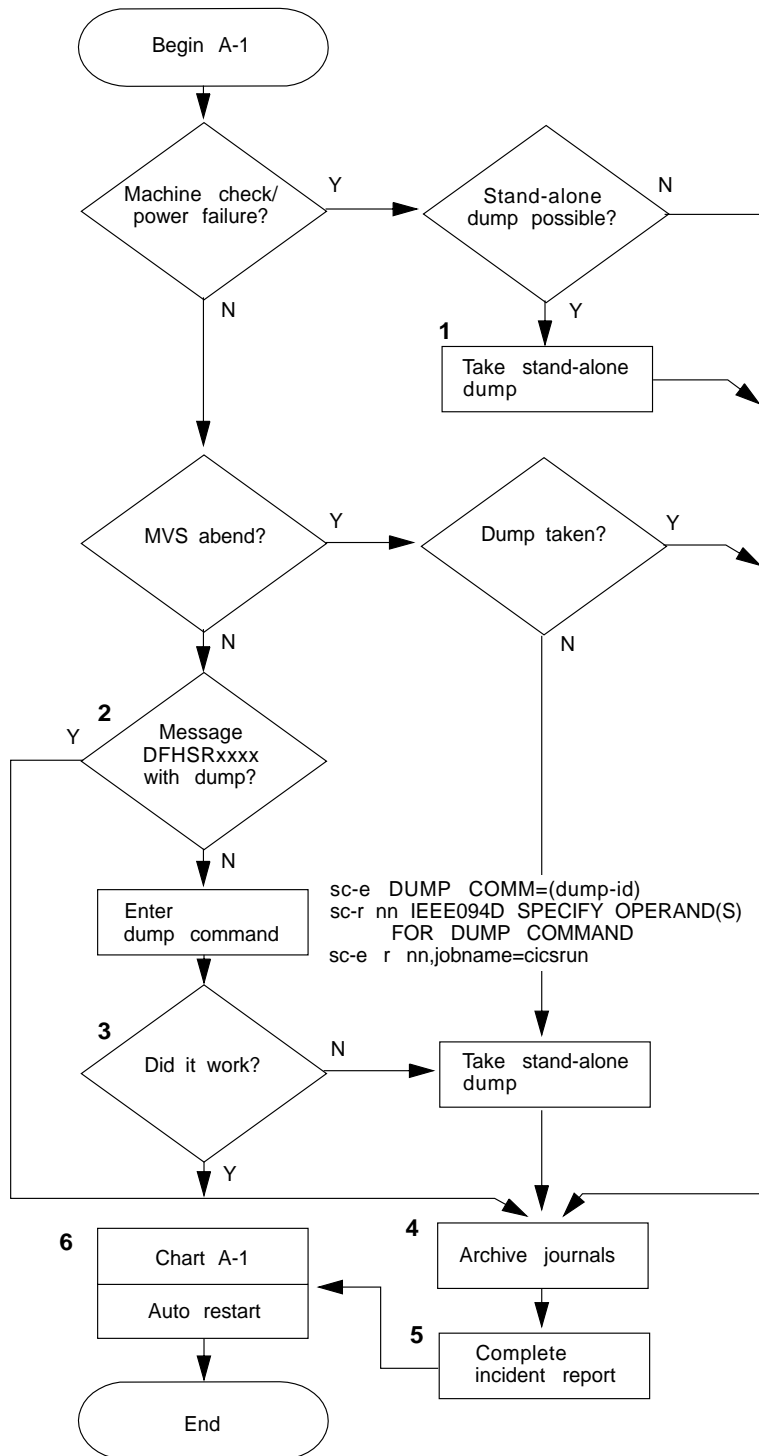


Figure 80. Chart A-1: Machine check, power failure, MVS loop, or abend

Notes for chart A-1

1. In the case of a machine check, a dump is useful for customer engineers and program service representatives to determine the cause of the problem.
2. Using the message DFHSRxxxx displayed on the operating system console, check in the *CICS/ESA Messages and Codes* manual to see whether the dump is provided on the system dump data set when CICS is abnormally terminated. You should print the SYS1.DUMPnn data set before starting the next CICS session.
3. The operating system message IEA911E COMPLETE DUMP ON SYS1.DUMPxx tells you that the dump is successfully written in the SYS1.DUMP data set.
4. You may want to archive to tape any crucial disk journals (for example, the CICS system log) at this point. If you do, make sure that the tapes are correctly labeled and entered on the online sessions log sheet. If you are using the CICS automatic archiving facility, this step is unnecessary, since CICS automatically submits the necessary archive jobs during the subsequent emergency restart.
5. The CICS incident report is used here for follow-up by DC Administration.
Complete the CICS online sessions log sheet with stop date, time, and your name. Make sure that the "allocated label" column is correctly and completely filled in. State in the "Comments/incidents" area that the next startup must be an emergency restart. The information on this CICS online sessions log sheet is used in the following CICS start up.
6. If a machine check or power failure occurs, you can perform an emergency restart (START=AUTO resulting in an emergency start) as soon as the machine is up again, unless a disk failure is suspected.

Chart A-2: Action on user problem

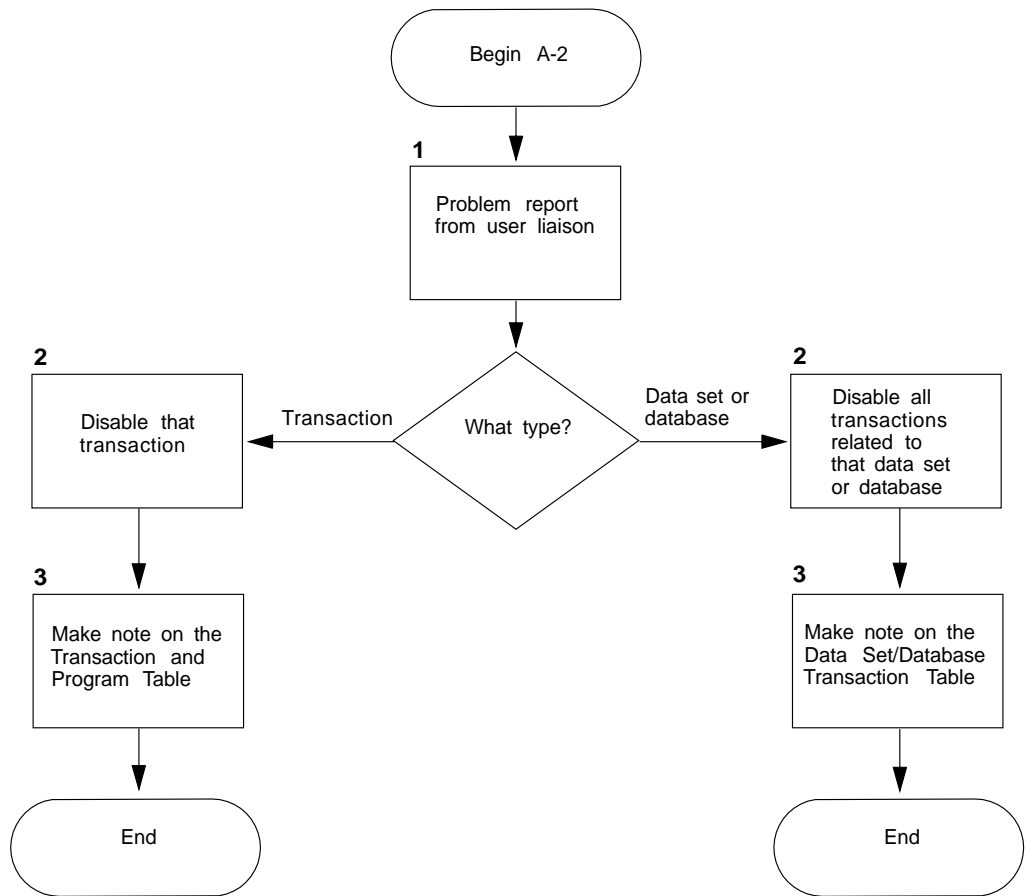


Figure 81. Chart A-2: Action on user problem

Notes for chart A-2

1. When a remote terminal operator or end user is experiencing difficulty, that person should contact user liaison. Your user liaison people can then determine the cause of the problem, and decide on the action to be taken. If the problem is related to transactions, data sets, or CICS local databases, user liaison may ask you to disable some transactions, by submitting a formal request. This request could be on the "Transaction and program table" for a transaction error, or on the "Data set/database and transaction table" for a data set or database error (see Figure 78 on page 296 and Figure 79 on page 298).

2. Disable the transaction(s), in accordance with the instructions received, by entering a CEMT SET TRANSACTION DISABLED command. Alternatively, depending on the complexity of the transactions that access the various databases or file control data sets, you might prefer to stop the databases or disable the CICS file control files. For details of CEMT SET master terminal commands, see the *CICS/ESA CICS-Supplied Transactions* manual.
3. After you have disabled the transaction(s), do not forget to complete the request by filling in the date, time, and your name on the form submitted to you.

Appendix B. CICS messages that cannot be edited

This appendix lists those messages that cannot be changed by the message editing utility.

DFHCC0101	DFHFB0123	DFHFB0505	DFHIR3706
DFHCC0201	DFHFB0124	DFHFB0506	DFHIR3707
DFHCC0300	DFHFB0125	DFHFB0550	DFHIR3708
DFHDB8199	DFHFB0126	DFHFB0551	DFHIR3709
DFHDB8297	DFHFB0128	DFHFB0552	DFHIR3710
DFHDU0006	DFHFB0129	DFHFB0553	DFHIR3711
DFHDU0201	DFHFB0130	DFHFB0554	DFHIR3712
DFHDU0202	DFHFB0131	DFHFB0555	DFHIR3713
DFHDU0205	DFHFB0132	DFHFB0556	DFHIR3714
DFHDU0208	DFHFB0133	DFHFB0557	DFHIR3715
DFHDU0209	DFHFB0134	DFHFB0558	DFHIR3716
DFHDU0210	DFHFB0150	DFHFB0559	DFHIR3717
DFHDU0211	DFHFB0151	DFHFB0560	DFHIR3718
DFHDU0309	DFHFB0152	DFHFB0561	DFHIR3719
DFHDU1601	DFHFB0153	DFHFB0600	DFHIR3720
DFHDU1602	DFHFB0154	DFHFB0601	DFHIR3721
DFHDU1603	DFHFB0155	DFHFB0602	DFHIR3722
DFHDU1604	DFHFB0156	DFHFB0603	DFHIR3723
DFHDU1609	DFHFB0157	DFHFB0604	DFHIR3724
DFHDX8300	DFHFB0200	DFHFB0605	DFHIR3725
DFHDX8301	DFHFB0201	DFHFB0606	DFHIR3726
DFHDX8302	DFHFB0202	DFHFB0607	DFHIR3727
DFHDX8303	DFHFB0203	DFHFB0608	DFHIR3728
DFHDX8304	DFHFB0204	DFHFB0609	DFHIR3729
DFHFB0010	DFHFB0205	DFHFB0610	DFHIR3730
DFHFB0011	DFHFB0206	DFHFB0611	DFHIR3731
DFHFB0100	DFHFB0210	DFHFB0612	DFHIR3732
DFHFB0101	DFHFB0211	DFHFB0613	DFHIR3733
DFHFB0102	DFHFB0212	DFHFB0614	DFHIR3734
DFHFB0103	DFHFB0400	DFHFB0615	DFHIR3735
DFHFB0105	DFHFB0401	DFHFB0616	DFHIR3736
DFHFB0106	DFHFB0450	DFHFB0621	DFHIR3737
DFHFB0107	DFHFB0451	DFHFB0622	DFHIR3738
DFHFB0108	DFHFB0452	DFHFB0623	DFHIR3739
DFHFB0109	DFHFB0453	DFHFB0624	DFHIR3740
DFHFB0110	DFHFB0454	DFHFB0650	DFHIR3741
DFHFB0111	DFHFB0455	DFHFB0651	DFHIR3742
DFHFB0112	DFHFB0456	DFHFB0701	DFHIR3743
DFHFB0113	DFHFB0457	DFHFB0702	DFHIR3744
DFHFB0114	DFHFB0458	DFHFB0703	DFHIR3745
DFHFB0116	DFHFB0459	DFHFB0704	DFHIR3746
DFHFB0117	DFHFB0460	DFHIR3700	DFHIR3765
DFHFB0118	DFHFB0461	DFHIR3701	DFHIR3795
DFHFB0119	DFHFB0500	DFHIR3702	DFHIR3797
DFHFB0120	DFHFB0501	DFHIR3703	DFHJC2900
DFHFB0121	DFHFB0502	DFHIR3704	DFHJC2901
DFHFB0122	DFHFB0503	DFHIR3705	DFHJC2902

DFHJC2903	DFHJC4586	DFHME0118	DFHMU0102
DFHJC2904	DFHJC4596	DFHME0119	DFHMU0103
DFHJC2905	DFHJC4597	DFHME0121	DFHMU0104
DFHJC2906	DFHJC4598	DFHME0122	DFHMU0105
DFHJC2907	DFHJC4599	DFHME0123	DFHMU0106
DFHJC2908	DFHJC6100	DFHME0124	DFHMU0107
DFHJC2909	DFHJC6101	DFHME0125	DFHMU0108
DFHJC2910	DFHJC6102	DFHME0126	DFHMU0109
DFHJC2911	DFHJC6103	DFHME0127	DFHMU0110
DFHJC2912	DFHJC6104	DFHME0128	DFHMU0111
DFHJC2913	DFHJC6105	DFHME0129	DFHMU0112
DFHJC2914	DFHJC6107	DFHME0130	DFHMU0113
DFHJC2915	DFHJC6110	DFHME0131	DFHMU0114
DFHJC2916	DFHJC6111	DFHME0132	DFHMU0115
DFHJC2917	DFHJC6199	DFHME0133	DFHMU0116
DFHJC2918	DFHKE0308	DFHME0134	DFHMU0117
DFHJC2919	DFHKE0005	DFHME0135	DFHMU0118
DFHJC2920	DFHKE0101	DFHME0136	DFHMU0119
DFHJC2921	DFHKE0102	DFHME0500	DFHMU0120
DFHJC2922	DFHKE0103	DFHME0501	DFHMU0121
DFHJC2923	DFHKE0104	DFHME0502	DFHMU0122
DFHJC2924	DFHKE0201	DFHME0503	DFHMU0123
DFHJC2925	DFHKE0202	DFHME0504	DFHMU0124
DFHJC2926	DFHKE0208	DFHME0505	DFHMU0125
DFHJC2927	DFHKE0209	DFHME0506	DFHMU0126
DFHJC2928	DFHKE0210	DFHME0507	DFHMU0127
DFHJC4502	DFHKE0301	DFHME0508	DFHMU0128
DFHJC4504	DFHKE0302	DFHME9993	DFHMU0129
DFHJC4505	DFHKE0303	DFHME9994	DFHMU0130
DFHJC4506	DFHKE0401	DFHME9996	DFHMU0131
DFHJC4507	DFHKE0407	DFHME9997	DFHMU0132
DFHJC4508	DFHKE0408	DFHME9998	DFHMU0133
DFHJC4522	DFHKE0999	DFHME9999	DFHMU0134
DFHJC4523	DFHKE1799	DFHMN0201	DFHMU0135
DFHJC4524	DFHKE1800	DFHMN0202	DFHMU0136
DFHJC4525	DFHMC4000	DFHMN0203	DFHMU0137
DFHJC4526	DFHME0001	DFHMN0204	DFHMU0138
DFHJC4527	DFHME0002	DFHMN0205	DFHMU0139
DFHJC4528	DFHME0004	DFHMN0206	DFHMU0140
DFHJC4529	DFHME0006	DFHMN0207	DFHMU0141
DFHJC4530	DFHME0101	DFHMN0208	DFHMU0142
DFHJC4531	DFHME0102	DFHMN0209	DFHMU0143
DFHJC4532	DFHME0105	DFHMN0210	DFHMU0144
DFHJC4533	DFHME0106	DFHMN0211	DFHMU0145
DFHJC4534	DFHME0107	DFHMN0212	DFHMU0146
DFHJC4559	DFHME0108	DFHMN0213	DFHMU0147
DFHJC4560	DFHME0109	DFHMN0214	DFHMU0148
DFHJC4561	DFHME0110	DFHMN0215	DFHMU0149
DFHJC4562	DFHME0111	DFHMN0216	DFHMU0150
DFHJC4563	DFHME0112	DFHMN0217	DFHMU0151
DFHJC4564	DFHME0113	DFHMN0218	DFHMU0152
DFHJC4565	DFHME0114	DFHMN0219	DFHMU0153
DFHJC4583	DFHME0115	DFHMN0220	DFHMU0154
DFHJC4584	DFHME0117	DFHMN0221	DFHMU0155

DFHMU0156	DFHDP0102	DFHXA6540	DFHXC6632
DFHMU0157	DFHDP0103	DFHXA6541	DFHXC6633
DFHMU0158	DFHDP0104	DFHXA6560	DFHXC6634
DFHMU0159	DFHDP0105	DFHXA6561	DFHXC6635
DFHMU0160	DFHDP0106	DFHXA6563	DFHXC6636
DFHMU0162	DFHDP0107	DFHXA6564	DFHXC6637
DFHMU0163	DFHDP0108	DFHXA6566	DFHXC6638
DFHMU0165	DFHDP0109	DFHXA6567	DFHXC6640
DFHMU0166	DFHDP0110	DFHXA6568	DFHXC6641
DFHMU0167	DFHDP0111	DFHXA6569	DFHXC6642
DFHMU0169	DFHDP0112	DFHXA6570	DFHXC6643
DFHMU0170	DFHDP0113	DFHXA6571	DFHXC6644
DFHMU0999	DFHDP0114	DFHXA6572	DFHXC6645
DFHMV0001	DFHDP0115	DFHXA6573	DFHXC6646
DFHPA1101	DFHDP0116	DFHXA6574	DFHXC6649
DFHPA1102	DFHDP0117	DFHXA6575	DFHXC6650
DFHPA1103	DFHDP0118	DFHXA6576	DFHXC6651
DFHPA1104	DFHDP0119	DFHXA6577	DFHXG6404
DFHPA1105	DFHDP0120	DFHXA6578	DFHXG6405
DFHPA1106	DFHDP0121	DFHXA6580	DFHXG6406
DFHPA1107	DFHDP0122	DFHXA6581	DFHXG6408
DFHPA1108	DFHDP0123	DFHXA6582	DFHXG6409
DFHPA1901	DFHDP0124	DFHXA6583	DFHXG6410
DFHPA1902	DFHDP0125	DFHXC6600	DFHXG6416
DFHPA1903	DFHDP0126	DFHXC6601	DFHXG6440
DFHPA1907	DFHDP0127	DFHXC6602	DFHXG6441
DFHPA1908	DFHDP0128	DFHXC6603	DFHXG6442
DFHPA1909	DFHDP0129	DFHXC6604	DFHXG6443
DFHPA1910	DFHDP0130	DFHXC6605	DFHXG6444
DFHPA1911	DFHSI1520	DFHXC6606	DFHXG6445
DFHPA1912	DFHTR0101	DFHXC6607	DFHXG6446
DFHPA1913	DFHTR0102	DFHXC6608	DFHXG6447
DFHPA1914	DFHTR0103	DFHXC6609	DFHXG6450
DFHPA1915	DFHTR0105	DFHXC6610	DFHXG6451
DFHPA1916	DFHTR0107	DFHXC6611	DFHXG6452
DFHPA1917	DFHTR0108	DFHXC6612	DFHXG6453
DFHPA1918	DFHTR0109	DFHXC6613	DFHXG6454
DFHPA1919	DFHTR0110	DFHXC6614	DFHXG6518
DFHPA1920	DFHTR0114	DFHXC6615	DFHXG6680
DFHPA1921	DFHTR0115	DFHXC6616	DFHXG6681
DFHPA1922	DFHTR0116	DFHXC6617	DFHXO6700
DFHPA1923	DFHTR2000	DFHXC6618	DFHXO6702
DFHPA1926	DFHTR2001	DFHXC6620	DFHXO6703
DFHPA1927	DFHTR2002	DFHXC6621	DFHXO6704
DFHPA1928	DFHTR2003	DFHXC6622	DFHXO6705
DFHPA1929	DFHTR2004	DFHXC6623	DFHXO6706
DFHPA1930	DFHTR3001	DFHXC6624	DFHXO6707
DFHPA1931	DFHTR3002	DFHXC6625	DFHXO6708
DFHPA1932	DFHTR3003	DFHXC6626	DFHXO6709
DFHPA1934	DFHTR3010	DFHXC6627	DFHXO6712
DFHPA1935	DFHTS1305	DFHXC6628	DFHXO6720
DFHPA1936	DFHTS1308	DFHXC6629	DFHXO6721
DFHPA1937	DFHTS1312	DFHXC6630	DFHXS1109
DFHDP0101	DFHXA6526	DFHXC6631	DFHZC0199

DFHZC2307	DFH5168	DFH5230	DFH5512
DFHZC2312	DFH5169	DFH5231	DFH5513
DFHZC2320	DFH5171	DFH5232	DFH5514
DFH0100	DFH5172	DFH5233	DFH5515
DFH0101	DFH5173	DFH5234	DFH5516
DFH0102	DFH5174	DFH5235	DFH5517
DFH0103	DFH5175	DFH5236	DFH5518
DFH0104	DFH5176	DFH5240	DFH5519
DFH0105	DFH5177	DFH5241	DFH5520
DFH4200	DFH5178	DFH5242	DFH5521
DFH5100	DFH5179	DFH5251	DFH5522
DFH5101	DFH5180	DFH5252	DFH5523
DFH5102	DFH5181	DFH5253	DFH5524
DFH5103	DFH5182	DFH5254	DFH5525
DFH5104	DFH5183	DFH5255	DFH5526
DFH5105	DFH5184	DFH5256	DFH5527
DFH5107	DFH5186	DFH5257	DFH5528
DFH5108	DFH5187	DFH5258	DFH5529
DFH5109	DFH5188	DFH5259	DFH5530
DFH5110	DFH5189	DFH5261	DFH5531
DFH5114	DFH5190	DFH5262	DFH5532
DFH5115	DFH5191	DFH5263	DFH5533
DFH5116	DFH5192	DFH5264	DFH5534
DFH5117	DFH5193	DFH5265	DFH5535
DFH5120	DFH5194	DFH5266	DFH5536
DFH5121	DFH5195	DFH5267	DFH5537
DFH5122	DFH5196	DFH5268	DFH5538
DFH5123	DFH5197	DFH5269	DFH5600
DFH5124	DFH5198	DFH5270	DFH5601
DFH5125	DFH5199	DFH5271	DFH5602
DFH5126	DFH5200	DFH5272	DFH5603
DFH5127	DFH5201	DFH5273	DFH5605
DFH5128	DFH5202	DFH5274	DFH5606
DFH5130	DFH5203	DFH5275	DFH5607
DFH5131	DFH5204	DFH5276	DFH5608
DFH5132	DFH5205	DFH5277	DFH5609
DFH5133	DFH5206	DFH5280	DFH5611
DFH5134	DFH5207	DFH5281	DFH5612
DFH5135	DFH5210	DFH5282	DFH5613
DFH5136	DFH5211	DFH5283	DFH5614
DFH5140	DFH5212	DFH5284	DFH5617
DFH5141	DFH5213	DFH5285	DFH5618
DFH5142	DFH5214	DFH5286	DFH5619
DFH5143	DFH5215	DFH5287	DFH5620
DFH5145	DFH5216	DFH5501	DFH5621
DFH5146	DFH5217	DFH5502	DFH5622
DFH5147	DFH5220	DFH5503	DFH5623
DFH5148	DFH5222	DFH5504	DFH5624
DFH5149	DFH5223	DFH5505	DFH5721
DFH5159	DFH5224	DFH5506	DFH5722
DFH5164	DFH5225	DFH5507	DFH5723
DFH5165	DFH5227	DFH5509	DFH5754
DFH5166	DFH5228	DFH5510	
DFH5167	DFH5229	DFH5511	

Index

Special Characters

@BCH, shared DL/I batch links 79

A

ADD command
 DFHCSDUP 223
ADD command, DFHCSDUP utility program 222
ALTER command
 DFHCSDUP 224
ALTER command, DFHCSDUP utility program 223
APPEND command
 DFHCSDUP 225, 226
 examples 226
APPEND command, DFHCSDUP utility program 224
authorized libraries 80, 119
automated operations 121
automatic journal archiving
 archive job submission, console message 52
 DFHJACDU, the automatic journal archiving utility 146
 in IMS data-sharing environment 58, 61
automatic start 18
auxiliary trace
 data sets 164
 using DFHTU410 to print 163
 using IPCS to print from GTF 169

B

batch region for shared DL/I databases 79

C

C= keyword, DFHJUP OPTION statement 139
cataloged procedures
 starting CICS as a batch job 18
 starting CICS as a started task 19
CBRC transaction 41, 61
CEBT INQUIRE AUXTRACE command 87
CEBT INQUIRE DUMP command 87
CEBT INQUIRE GTFTRACE command 87
CEBT INQUIRE INTTRACE command 87
CEBT INQUIRE SURVEILLANCE command 87
CEBT INQUIRE TAKEOVER command 86
CEBT PERFORM SHUTDOWN command 86
CEBT PERFORM SNAP command 87
CEBT PERFORM TAKEOVER command 86
CEBT SET AUXTRACE command 87
CEBT SET DUMP command 87
CEBT SET GTFTRACE command 87

CEBT SET INTTRACE command 87
CEBT SET SURVEILLANCE command 87
CEBT SET TAKEOVER command 86
CEBT, master terminal transaction 86
CEMT PERFORM SHUTDOWN command 85
CEMT, master terminal transaction 12, 41, 85, 178
CICS system definition file (CSD)
 offline utility program, DFHCSDUP 215
CICS410.SDFHAUTH, CICS load library 80, 119
CICSDATA, changing to new IPCS verb names 181
CLIST, TSO command list 121, 253
CLT (command list table)
 issuing commands from 103, 106
 write-to-operator messages 106
command list table (CLT)
 issuing commands from 103, 106
 write-to-operator messages 106
COND= keyword, DFHJUP OPTION statement 139
console devices
 using TSO command lists 121
console support under MVS/ESA SP 4.1 or later
 communicating with CICS 121
consoles
 entering system initialization parameters 21
CONTROL statement, DFHJUP utility program 136
control statements for DFHDU410 174
control statements, DFHTU410
 See DFHTU410 trace utility program, selection parameters
COPY command
 DFHCSDUP 227, 228
 examples 228
 FROMCSD option 227
 MERGE option 227
 REPLACE option 227
COPY command, DFHCSDUP utility program 226
COPY keyword, DFHJUP OPTION statement 138
CSD (CICS system definition file)
 offline utility program, DFHCSDUP 215

D

D= keyword, DFHJUP CONTROL statement 136
D= keyword, DFHJUP OPTION statement 140
data set/database and transaction table 295
data sharing in an IMS environment
 See IMS data sharing
database recovery control (DBRC) 31
 CBRC, CICS master terminal transaction 61
DBRC (database recovery control) 31
 CBRC, CICS master terminal transaction 61

DBRC commands, issuing 61
 DDNAME= keyword, DFHJUP CONTROL statement 136
 DDNAME= keyword, DFHJUP OPTION statement 140
 DDNOUT= keyword, DFHJUP CONTROL statement 137
 DEFINE command
 DFHCSDUP 229
 examples 229
 DEFINE command, DFHCSDUP utility program 228
 DELETE command
 DFHCSDUP 231
 examples 231
 DELETE command, DFHCSDUP utility program 230
 departmental titles 291
 DFH\$INDB utility program, to identify in-doubt resource updates 249
 DFH\$MOLS sample print program 197
 control statements 205, 206
 BREAK 206
 comments 206
 DATE 208
 IGNORE 209
 PRINT 210
 SELECT 210
 SORT 211
 monitoring sample utility 202
 overview 202
 sample job stream for 204
 DFH\$STED, utility to stagger the end-of-day time 255
 DFHAUXT auxiliary trace data set 164
 DFHBUXT auxiliary trace data set 164
 DFHCSD CICS system definition file
 See CSD (CICS system definition file)
 DFHCSDUP system definition utility program 215
 ADD command 222
 ALTER command 223
 APPEND command 224
 command processing considerations 222
 COPY command 226
 DEFINE command 228
 DELETE command 230
 EXTRACT command 232
 INITIALIZE command 234
 invocation from a user program 219
 invoking as a batch program 217
 LIST command 235
 MIGRATE command 237
 processing system definition file 215
 REMOVE command 240
 SERVICE command 241
 UPGRADE command 242
 VERIFY command 243
 DFHCSVC, the CICS type 3 SVC 23
 for DL/I shared databases 79
 DFHDRP program, CICS-DL/I batch-region controller 79
 coding EXEC statement 81
 DFHDU410 transaction dump utility program
 INDEX DD statement for dump summary 173
 job control statements for 177
 processing transaction dump data sets 173
 SYSIN control statements 174
 DFHFTAP utility program, tape formatting 133
 DFHIPCSP, IPCS exit control table member for CICS 180
 DFHJACDU journal archive utility program 146
 DFHJCJFP utility program, disk data set formatting 133
 DFHJUP journal utility program
 See journal utility program, DFHJUP
 DFHKE1799 message 15, 16, 127, 128
 DFHMNDUP dictionary utility program 197
 DFHMSCAN utility program, to identify macro-level programs
 detailed report 247
 how the program works 246
 JCL 245
 limitations of 248
 overview 245
 summary report 246
 DFHOSD data set 116
 DFHPD410, SDUMP formatting exit routine for IPCS 181
 sample jobs for formatting CICS SDUMPs 192
 DFHPDX, SDUMP formatting exit routine of IPCS 180
 DFHSMUTL local catalog utility program. 287
 DFHSNMIG utility program, to migrate from SNT to RACF database 253
 sample job stream to execute CLIST 254
 sample job stream to run utility program 254
 DFHSTUP statistics utility program
 control parameters 156
 job stream for 153
 printing CICS statistics 151
 DFHTEOF utility program, tape end-of-file 133
 DFHTU410 trace utility program
 printing auxiliary trace data 163
 selection parameters
 ABBREV 165
 ALL 165
 ENTRY_NUM 165
 EXCEPTION 165
 FULL 165
 KE_NUM 165
 PAGESIZE 164, 166
 TASKID 166
 TERMID 166
 TIMERG 166
 TRANID 166
 TYPETR 167

DISPLAY command for XRF overseer 112
 DL/I batch links, shared 79
 DL/I shared database operation
 IRC links for 79
 MVS checkpointing 80
 shared DB region 79
 dump exit routines 181, 182
 sample jobs for formatting CICS SDUMPs 192
 dumps 70
 CICS transaction dumps 173
 DFHDU410 utility program, to process CICS
 transaction dumps 173
 DFHIPCSP, exit control table member for CICS 180
 DFHPD410, SDUMP formatting exit routine of
 IPCS 181
 dump summary 192
 example 192
 error index 192
 exit parameters for CICS 182, 183
 exit routines for IPCS 181, 182
 formatting with IPCS 182
 IPCS exit control table 180
 IPCS verb names for CICS 181
 sample formatting jobs using IPCS 192
 SDUMP formatting exit routines of IPCS 180
 SDUMP options needed for the CICS dump
 exit 180
 stand-alone dump (SADMP) 70
 VERBEXIT command of IPCS 182
 via MVS SDUMP macro 173
 processing using IPCS 179
 What to do before formatting with IPCS 179

E

E= keyword, DFHJUP OPTION statement 140
 editing messages 257
 emergency restart 74
 END command, to terminate the overseer 115
 error procedures
 A-1 machine check, power failure, MVS loop, or
 abend 300
 A-2 action on user problem 302
 EXEC CICS PERFORM SHUTDOWN command 13,
 127
 EXEC CICS PERFORM SHUTDOWN IMMEDIATE
 command 13
 EXITR= keyword, DFHJUP OPTION statement 140
 extended recovery facility (XRF)
 authorization for DFHCRST 115
 batch region job for the CICS overseer 115
 CEBT command 85
 CEMT command 85
 CEMT PERFORM SHUTDOWN command 85
 clock synchronization (multi-MVS environment) 88
 DFH\$AXRO job, sample overseer 115

extended recovery facility (XRF) (*continued*)
 DFHOSD data set 116
 monitoring the CICS regions 112
 operations 83
 overseer 112
 overseer, sample job 115
 ready region, IMS data-sharing 37
 restarting regions in place 114
 shutting down the active 91
 shutting down the alternate CICS 92
 starting an active 88
 starting an alternate CICS 89
 starting CICS with XRF 88
 takeover 94, 95, 97
 abnormal sign-off by the active 106
 after a command 103
 CEBT PERFORM TAKEOVER 103
 checking for end of the active 98
 completing 103
 events that can cause 95
 takeover after MVS failure 111
 what happens during 97
 when it occurs 97
 when surveillance signals cease 108
 EXTRACT command (DFHCSDUP)
 examples 233
 OBJECTS option 233
 USERPROGRAM option 233
 EXTRACT command, DFHCSDUP utility program 232

F

FEPI
 See front end programming interface
 FLDLEN= keyword, DFHJUP OPTION statement 139
 FLDTYP= keyword, DFHJUP OPTION statement 138
 flowcharts for error procedures
 A-1 machine check, power failure, MVS loop, or
 abend 300
 A-2 action on user problem 302
 forms and tables for operating procedures 291
 CICS incident report 293
 CICS online sessions log sheet 291
 data set/database and transaction table 295
 transaction and program table 297
 front end programming interface (FEPI)
 dump control option 189

G

generalized trace facility (GTF)
 CEBT GTFTRACE command 87
 formatting trace records 170
 GTFTRACE subcommand of IPCS 170
 printing CICS traces using IPCS 169
 sample job to print GTF trace entries 171

GTF (generalized trace facility)
See generalized trace facility (GTF)

H

H= keyword, DFHJUP CONTROL statement 136

I

IMS data sharing 29
 CICS startup and shutdown 43
 operations 37
 SIT parameter DLDBRC 34
 system initialization parameter DLIRLM 34
IMS flowcharts, operating procedures 42
IMS operating procedures and flowcharts
 how to use 42
 index (IMS data-sharing) 43
 overview (IMS data-sharing) 42
 stand-alone dump (SADMP) 70
 start up CICS 45
 taking system dump 70
IMS resource lock manager (IRLM)
 ACTCOMM command 67
 ACTCOMM command 56, 64, 65, 67, 68
 connection failure 56, 64
 local failure 56
 re-establishing connection 64
 ready region 37
 reconnecting CICS subsystem after IRLM restart 56
 restarting after failure 56
 SETSTATE command 65, 67
 STATUS command 65
 stopping 53
in-doubt resolution program 249
incident report form 293
INITIALIZE command (DFHCSDUP) 235
INITIALIZE command, DFHCSDUP utility program 234
initializing the network 43
interactive problem control system (IPCS)
 DFHIPCSP, IPCS exit control table member for
 CICS 180
 dump summary 192
 example 192
 error index 192
 IPCS verb names for CICS 181
 printing CICS traces from GTF 169
 processing CICS SDUMPs 179
 sample formatting jobs 192
 SDUMP options needed for the CICS dump
 exit 180
interregion communication (IRC) 26
IPCS (interactive problem control system)
 See interactive problem control system (IPCS)
IRC (interregion communication) 26

IRC links for CICS shared database
 IRC @BCH definitions in the CSD 79
IRCSTRT, system initialization parameter 79
IRLM (IMS resource lock manager)
 See IMS resource lock manager (IRLM)
IRLM ready region 37
IRLM, specifying the name 34

J

JACD (journal archive control data set) 146
JCL to submit CICS commands 122
job control language (JCL)
 for CICS as a batch job 18
 for CICS as a started task 19
job titles 291
journal archive control data set (JACD) 146
journal automatic archiving
 archive job submission, console message 52
 DFHJACDU, the automatic journal archiving
 utility 146
 in IMS data-sharing environment 58, 61
journal data sets 45
journal utility program, DFHJUP
 control statements 135
 COMMENTS 141
 CONTROL 136
 END 141
 examples of 141
 OPTION 137
 description of 133
 input and output 133
 JCL requirements 134

K

K= keyword, DFHJUP CONTROL statement 136
keywords
 C=, of DFHJUP OPTION statement 139
 COND=, of DFHJUP OPTION statement 139
 COPY, of DFHJUP OPTION statement 138
 D=, of DFHJUP CONTROL statement 136
 D=, of DFHJUP OPTION statement 140
 DDNAME=, of DFHJUP CONTROL statement 136
 DDNAME=, of DFHJUP OPTION statement 140
 DDNOUT=, of DFHJUP CONTROL statement 137
 E=, of DFHJUP OPTION statement 140
 EXITR=, of DFHJUP OPTION statement 140
 FLDLN=, of DFHJUP OPTION statement 139
 FLDTYP=, of DFHJUP OPTION statement 138
 H=, of DFHJUP CONTROL statement 136
 K=, of DFHJUP CONTROL statement 136
 L=, of DFHJUP OPTION statement 139
 O=, of DFHJUP CONTROL statement 137
 O=, of DFHJUP OPTION statement 138
 OFFSET=, of DFHJUP OPTION statement 138

keywords (*continued*)

- P=, of DFHJUP OPTION statement 140
- PRINT, of DFHJUP OPTION statement 138
- PRTSYS=, of DFHJUP OPTION statement 140
- SKIP=, of DFHJUP CONTROL statement 136
- STOPAFT=, of DFHJUP CONTROL statement 136
- T=, DFHJUP OPTION statement 138
- V=, of DFHJUP OPTION statement 138
- VALUE=, of DFHJUP OPTION statement 138

L

- L= keyword, DFHJUP OPTION statement 139
- libraries, authorized 80, 119
- LIST command
 - DFHCSDUP 236
- LIST command (DFHCSDUP)
 - OBJECTS option 236
- LIST command, DFHCSDUP utility program 235
- local catalog utility program, DFHSMUTL 287
- log sheet, online sessions 293

M

- machine check 13
- macro-level programs, identifying 245
- master terminal transaction, CEMT 12
- message editing utility 257
 - defining the utility data set index 258
 - edit message panel 267
 - help panels 274
 - installing 257
 - language selection panel 262
 - main panel 264
 - message edit panel 271
 - message edit selection panel 266
 - PTF update panel 269
 - requirements 257
 - set defaults panels 260
 - using 259
 - applying PTFS 269
 - applying PTFS, rules 270
 - applying PTFS, sample output 270
 - copying message data set members 265
 - editing messages 267
 - editing messages, rules to be observed 271
 - generating message load modules 268
 - getting help 274
 - link editing changed message source members 268
 - performing actions on message data sets 264
 - selecting languages for translation 262
 - selecting message sets to be edited 266
 - sorting lists of message set members 268
 - specifying default values 260
 - starting 259

messages

- replying to messages 124
- replying to messages from transactions 125
- suppressing 124
- messages, suppressing and rerouting 126
- MIGRATE command
 - DFHCSDUP 238
 - TABLE option 238
- MIGRATE command (DFHCSDUP)
 - examples 240
 - FCT migration 238
 - TCT migration 238, 239
 - TYPESGROUP option 238
 - TYPETERM creation 239
- MIGRATE command, DFHCSDUP utility program 237
- migration utility program, SNT to RACF 253
- MODIFY command 122
- monitoring
 - CICS regions 112
 - dictionary utility program, DFHMNDUP 197
 - sample print program, DFH\$MOLS 197
- MRO (multiregion operation) 26
- multiregion operation (MRO) 26
- MVS checkpointing, for CICS shared databases 80
- MVS START command, to start CICS 19

N

- NEGOFF keyword, DFHJUP OPTION statement 138

O

- O= keyword, DFHJUP CONTROL statement 137
- O= keyword, DFHJUP OPTION statement 138
- OBJECTS option
 - LIST command (DFHCSDUP) 236
- OFFSET= keyword, DFHJUP OPTION statement 138
- online sessions log sheet 293
- operating system failure 13
- operations, automated 121
- operator communication for initialization parameters 21
- overriding system initialization parameters
 - from the console 21
- overseer user program
 - DFHOSD data set 116
 - displaying states of CICS regions 112
 - how it sees restart-in-place function 114
 - job control statements 115
 - operating 112
 - restarting CICS regions in place 114
 - sample output from the overseer 120
 - taking a snap dump 115
 - terminating 115

P

P= keyword, DFHJUP OPTION statement 140
power failure 13
PRINT keyword, DFHJUP OPTION statement 138
program and transaction table 297
PRTSYS= keyword, DFHJUP OPTION statement 140
PTFs, DFHMEU update log (sample output) 270

R

RACF (resource access control facility)
library security 119
migrating entries from the CICS SNT 253
RDI job, IRLM ready region 37
ready region, IRLM 37
regions, restarting in place 114
REMOVE command
DFHCSDUP 241
examples 241
REMOVE command, DFHCSDUP utility program 240
REPLY command, responding to console messages 125
resource access control facility (RACF)
library security 119
migrating entries from the CICS SNT 253
restart, emergency 74
restarting CICS regions in place 114

S

SADMP, MVS/ESA stand-alone dump 70
sample resource manager, in-doubt resolution 249
SERVICE command (DFHCSDUP)
FROMCSD operand 242
LEVEL operand 242
SERVICE command, DFHCSDUP utility program 241
shared DB region 79
running batch region for DL/I shared databases 79
shared DL/I batch links 79
shutdown 51
SKIP= keyword, DFHJUP CONTROL statement 136
SNAP command, to take a snap dump of the overseer 115
SNT to RACF migration utility program 253
staggering the end-of-day time 255
stand-alone dump 70
START command, MVS 19
START, system initialization parameter
START=AUTO 18
START=COLD 18
START=LOGTERM 18
START=STANDBY 18
started task, CICS as a 19
starting CICS regions 3, 17
as a batch job 18
as a started task 19

starting CICS regions (*continued*)
MVS START command 19
START=AUTO 18
START=STANDBY, for an XRF alternate CICS 18
statistics utility program, DFHSTUP 151
statistics, staggering the end-of-day time 255
STOPAFT= keyword, DFHJUP CONTROL statement 136
STORECLOCK value warm start 8
SVC
specifying DFHC SVC in the startup job 23
type 3, for DL/I shared databases 79
SVC type 3, for DL/I shared databases 79
system
control 11
system initialization
for an alternate CICS (XRF=YES)
START=STANDBY 18
START=AUTO 18
system initialization parameters
entering at the console 21
from operator's console 21
system startup 3, 17
system task 19

T

T= keyword, DFHJUP OPTION statement 138
time sharing option (TSO)
the DFHCSDUP program 219
using command lists 121
trace
using DFHTU410 to print 163
using IPCS to print from GTF 169
utility programs 163
trace utility program, DFHTU410 163
transaction and program table 297
transaction list table (XLT) 14
translating messages 257
TSO (time sharing option)
the DFHCSDUP program 219
using command lists 121
tuning the system 11
type 3 SVC, DL/I shared DB requirement 79

U

unit of recovery descriptor (URD) 8
UPGRADE command (DFHCSDUP)
USING operand 243
UPGRADE command, DFHCSDUP utility program 242
URD (unit of recovery descriptor) 8
user programs, sample overseer 112
utility programs 13
utility programs, offline
identifying in-doubt resource updates (DFH\$INDB) 249

utility programs, offline (*continued*)
 identifying macro-level programs (DFHMSCAN) 245
 initializing CICS system definition file,
 DFHCSDUP 215
 migrating SNT entries to the RACF database
 (DFHSNMIG) 253
 preparing statistics reports, DFHSTUP 151
 preparing trace reports, DFHTU410 163
 processing transaction dump data sets
 (DFHDU410) 173
 running under TSO 219
 staggering the end-of-day time (DFH\$STED) 255

XRF (extended recovery facility) (*continued*)
 takeover (*continued*)
 events that can cause 95
 takeover after MVS failure 111
 what happens during 97
 when it occurs 97
 when surveillance signals cease 108

V

V= keyword, DFHJUP OPTION statement 138
VALUE= keyword, DFHJUP OPTION statement 138
VERBEXIT subcommand of IPCS 182
 exit parameters for CICS 182, 183
VERBEXIT, IPCS subcommand 179
VERIFY command (DFHCSDUP) 244
VERIFY command, DFHCSDUP utility program 243

W

warm-start-possible indicator 15

X

XLT (transaction list table) 14
XMEOUT, global exit for message handling 126
XRF (extended recovery facility)
 authorization for DFHCRST 115
 batch region job for the CICS overseer 115
 CEBT command 85
 CEMT command 85
 CEMT PERFORM SHUTDOWN command 85
 clock synchronization (multi-MVS environment) 88
 DFH\$AXRO job, sample overseer 115
 DFHOSD data set 116
 monitoring the CICS regions 112
 operations 83
 overseer 112
 overseer, sample job 115
 ready region, IMS data-sharing 37
 restarting regions in place 114
 shutting down the active 91
 shutting down the alternate CICS 92
 starting an active 88
 starting an alternate CICS 89
 starting CICS with XRF 88
 takeover 94, 95, 97
 abnormal sign-off by the active 106
 after a command 103
 CEBT PERFORM TAKEOVER 103
 checking for end of the active 98
 completing 103

Sending your comments to IBM

CICS for MVS/ESA

Operations and Utilities Guide

SC33-1167-02

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM.

Feel free to comment on what you regard as specific errors or omissions, and on the accuracy, organization, subject matter, or completeness of this book. Please limit your comments to the information in this book and the way in which the information is presented.

To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate, without incurring any obligation to you.

You can send your comments to IBM in any of the following ways:

- By mail, use the Readers' Comment Form
- By fax:
 - From outside the U.K., after your international access code use 44 962 870229 (after 16 April 1995, use 44 1962 870229)
 - From within the U.K., use 0962 870229 (after 16 April 1995, use 01962 870229)
- Electronically, use the appropriate network ID:
 - IBM Mail Exchange: GBIBM2Q9 at IBMAIL
 - IBMLink: WINVMJ(IDRCF)
 - Internet: idrcf@winvmj.vnet.ibm.com

Whichever you use, ensure that you include:

- The publication number and title
- The page number or topic to which your comment applies
- Your name and address/telephone number/fax number/network ID.

Readers' Comments

CICS for MVS/ESA

Operations and Utilities Guide

SC33-1167-02

Use this form to tell us what you think about this manual. If you have found errors in it, or if you want to express your opinion about it (such as organization, subject matter, appearance) or make suggestions for improvement, this is the form to use.

To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer. This form is provided for comments about the information in this manual and the way it is presented.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Be sure to print your name and address below if you would like a reply.

Name

Address

Company or Organization

Telephone

Email



CICS/ESA Operations and Utilities Guide
SC33-1167-02

You can send your comments POST FREE on this form from any one of these countries:

Australia	Finland	Iceland	Netherlands	Singapore	United States
Belgium	France	Israel	New Zealand	Spain	of America
Bermuda	Germany	Italy	Norway	Sweden	
Cyprus	Greece	Luxembourg	Portugal	Switzerland	
Denmark	Hong Kong	Monaco	Republic of Ireland	United Arab Emirates	

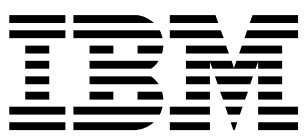
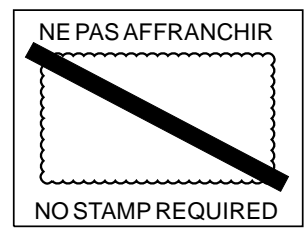
If your country is not listed here, your local IBM representative will be pleased to forward your comments to us. Or you can pay the postage and send the form direct to IBM (this includes mailing in the U.K.).

1 Cut along this line

2 Fold along this line

By air mail
Par avion

IBRS/CCRINUMBER: PHQ - D/1348/SO



**REPONSE PAYEE
GRANDE-BRETAGNE**

IBM United Kingdom Laboratories
Information Development Department (MP095)
Hursley Park,
WINCHESTER, Hants
SO21 2ZZ United Kingdom

3 Fold along this line

From: Name _____
Company or Organization _____
Address _____

EMAIL _____
Telephone _____

1 Cut along this line

4 Fasten here with adhesive tape



Program Number: 5655-018

Printed in U.S.A.

SC33-1167-02



Spine information:



CICS for MVS/ESA

Operations and Utilities Guide

Version 4 Release 1

SC33-1167-02

Artwork Definitions

<u>id</u>	<u>File</u>	<u>Page</u>	<u>References</u>
CESAINS	DFHA6SET	i	

Definition List (DL) Definitions

<u>id</u>	<u>File</u>	<u>Page</u>	<u>References</u>
KEYWDS	DFHA6010	185	185, 185, 186, 186, 186, 186, 186, 186, 186, 186, 186, 186, 187, 187, 187, 187, 187, 187, 187, 187, 188, 188, 188, 188, 188, 188, 188, 188, 188, 188, 189, 189, 189, 189, 189, 189, 190, 190, 190, 191, 191, 191, 191, 191

Figure Definitions

<u>id</u>	<u>File</u>	<u>Page</u>	<u>References</u>
BOXF	DFHA6SET	i	19, 37, 117, 118, 120, 148, 153, 164, 168, 171, 174, 177, 182, 192, 193, 200, 204, 217, 221, 254, 254
DBCDEF	DFHA6SET	i	
IVPFDEF	DFHA6SET	i	
JOBFIG	DFHA6SET	i	22
JOBCFIG	DFHA6SET	i	250, 251, 252, 279, 280, 280

LERS Definitions

<u>id</u>	<u>File</u>	<u>Page</u>	<u>References</u>
FAMLEDI	DFHA6SET	i	
FAMLEDS	DFHA6SET	i	

Table Definitions

<u>id</u>	<u>File</u>	<u>Page</u>	<u>References</u>
RULES	DFHA6SET	i	116
STD	DFHA6SET	i	
NORULES	DFHA6SET	i	43
BOXT	DFHA6SET	i	
HDRULES	DFHA6SET	i	33
SYNTAX	DFHA6SET	i	i, 136, 137, 141, 141
SITSYNT	DFHA6SET	i	i
PADFT	DFHA6SET	i	
PADFOOT	DFHA6SET	i	
TROW	DFHL99L2	xii	xii, xii, xii, xii, xii, xii, xii
LIBXB1	DFHL99L2	xii	xii, xii
PAD	DFHL99L2		

HEAD	DFHA6BIB	xii	xii, xii, xii, xii, xii, xii
OPVAL2	DFHA6007	xvii	xvii
OPVAL3	DFHA6007	145	145, 146
FSTUPSI	DFHA6008	145	145
		157	157

Syntax Definitions

<u>id</u>	<u>File</u>	<u>Page</u>	<u>References</u>
SYNSYN	DFHA6SET	i	223, 224, 225, 227, 229, 231, 233, 235, 236, 238, 241, 242, 243, 244

Example Definitions

<u>id</u>	<u>File</u>	<u>Page</u>	<u>References</u>
LBLBOX	DFHA6SET	i	

Figures

<u>id</u>	<u>File</u>	<u>Page</u>	<u>References</u>
STCICSJ	DFHA6STR	19	1 18
STCICS	DFHA6STR	20	2 20, 22
TSTART	DFHA6CPM	22	3 22, 22, 88
DSTAB1	DFHA6002	35	4 34, 34
DSTAB2	DFHA6002	35	5
DSTAB3	DFHA6002	36	6
DSTAB4	DFHA6002	36	7 34, 34
RDIRGN	DFHA6002	38	8 37
DFSUARC	DFHA6002	40	9 39
Q1	DFHA6SXR	93	19 91
Q2	DFHA6SXR	94	20 92
ARROWS	DFHA6SXR	96	21 96
VAL41LP	DFHA6SXR	100	22 99
VAL41	DFHA6SXR	101	23 100
R1	DFHA6SXR	104	24 103
R3	DFHA6SXR		

		107	25	106
R2	DFHA6SXR			
		109	26	108
OSDJOB	DFHA6SXR			
		117	27	116, 117, 119, 119
XRFOVSR	DFHA6SXR			
		118	28	118
ORT	DFHA6SXR			
		120	29	120
JUPEG1	DFHA6007			
		142	30	141
JUPEG2	DFHA6007			
		142	31	142
JACDU	DFHA6007			
		149	32	148
STUPJOB	DFHA6008			
		153	33	153, 154
TRCJOB	DFHA6009			
		164	34	164
TRFIG	DFHA6009			
		169	35	167, 167
GTFJOB	DFHA6009			
		171	36	170, 171, 171
DUPKEYW	DFHA6010			
		174	37	
DUPJOB	DFHA6010			
		177	38	177
IPCSP	DFHA6010			
		181	39	180, 181, 182
PDXPARM	DFHA6010			
		182	40	182
SUMFIG	DFHA6010			
		192	41	192
IPCSDIR	DFHA6010			
		193	42	192
IPCSJOB	DFHA6010			
		194	43	172, 192, 193, 194
MNDUP	DFHA6011			
		201	44	200
MOLSJOB	DFHA6011			
		204	45	204
CEDAOU	DFHA6012			
		216	46	216, 216
CSDUP	DFHA6012			
		217	47	216, 217
CSDUPC	DFHA6012			
		222	48	221
COPFIG	DFHA4COP			
		228	49	228
DEFF	DFHA4DEF			
		230	50	
DELCSD	DFHA4DEL			
		232	51	
EXTCSD	DFHA4EXT			
		234	52	
MIGCSD	DFHA4MIG			
		240	53	

REMSCD	DFHA4REM			
MSCANRP	DFHA6013	241	54	
		247	55	
INDBASS	DFHA6014			247
		250	56	
INDBRUN	DFHA6014			250
		251	57	
INDBEGO	DFHA6014			251
		252	58	
SNMFIG	DFHA6015			252
		254	59	
CLISTB	DFHA6015			254
MEDFLT	DFHE4ICD	254	60	
		261	61	
MEDFT2	DFHE4ICD			260, 265
		262	62	
LNGSEL	DFHE4ICD			260
		264	63	
MEMAINS	DFHE4ICD			261, 262
		265	64	
MSGSEL	DFHE4ICD			264
		267	65	
MSGED	DFHE4ICD			266
		267	66	
PTFUP	DFHE4ICD			267, 267
		269	67	
MEUPTU	DFHE4ICD			265, 269
		271	68	
MSGINS	DFHE4ICD			270
		272	69	
MSGOPT	DFHE4ICD			272
		273	70	
MSGREP	DFHE4ICD			272
		273	71	
HELPO	DFHE4ICD			273
		275	72	
ASMIWUP	DFHE4ICY			
		279	73	
RUNIWUP	DFHE4ICZ			279
		280	74	
OUTIWUP	DFHE4IDA			280
		281	75	
M2T1	DFHA6001			280
		292	76	
M2T2	DFHA6001			
		294	77	
M2T3	DFHA6001			
		296	78	
M2T4	DFHA6001			302
		298	79	
A6ACTUP	DFHA6001			302
		302	81	

Headings

<u>id</u>	<u>File</u>	<u>Page</u>	<u>References</u>
SPECN	DFHL99SP	vii	Notices ii
TM	DFHL99TM	vii	Trademarks and service marks
A6TERM	DFHA6PRE	x	Notes on terminology 26
CURR	DFHQ99CU	xi	Determining if a publication is current xiii
LIBXB1	DFHL99L2	xii	CICS/ESA 4.1 library ix
PART1	DFHA6PT1	1	Part 1, Operating CICS regions xi
DFHA6OV	DFHA6OVW	3	Chapter 1, Overview of CICS operations xix, 1
STARTO	DFHA6OVW	3	Starting up CICS 17
STYPE	DFHA6OVW	4	The types of CICS startup
COLD	DFHA6OVW	5	CICS actions on a cold start 4
WARM	DFHA6OVW	5	CICS actions on a warm start 4
EMER	DFHA6OVW	8	CICS actions on an emergency restart 4
DFHG50A	DFHA6OVW	9	CICS-VTAM actions after an emergency restart
DFHA23L	DFHA6OVW	10	CICS startup and the VTAM session
DFHA2SZ	DFHA6OVW	11	End of CICS startup
DFHE30G	DFHA6OVW	11	Controlling CICS operation
DFHE30H	DFHA6OVW	13	CICS-supplied utility programs
SHUTD	DFHA6OVW	13	Shutting down CICS 127
STARTH	DFHA6STR	17	Chapter 2, Starting up CICS regions xx, 1, 3
A6SIPS0	DFHA6STR	17	Specifying system initialization parameters before startup 17
A6BTCH	DFHA6STR	18	Starting CICS as a batch job 17
DFHA23W	DFHA6STR	19	Starting CICS as a started task 17
OVRSIPS	DFHA6STR	21	Overriding system initialization parameters during startup 17
DFHA61S	DFHA6CPM	22	System console messages for CICS startup 126
DFHA61K	DFHA6005	25	Chapter 3, Operating CICS in a multiregion environment 1
DFHO102	DFHA6002	29	Chapter 4, CICS in an IMS data-sharing environment 1
DFHA606	DFHA6002	29	An overview of IMS data-sharing 29
DFHA607	DFHA6002	31	The levels of IMS data-sharing 29

DFHA608	DFHA6002	31	Database recovery control (DBRC) 29
DFHA609	DFHA6002	32	Access authorization 30
DFHA60A	DFHA6002	34	Internal resource lock manager (IRLM) 29
DFHA60B	DFHA6002	34	Coding CICS system initialization parameters for IMS data-sharing 29
DFHA60C	DFHA6002	34	Sample IMS data-sharing configurations 29
DFHA60D	DFHA6002	37	Overview of operating CICS in an IMS data-sharing environment 29
DISPSHR	DFHA6002	41	Database integrity
DFHA60F	DFHA6002	42	Operating procedures and flowcharts 29, 37
DFHA60G	DFHA6002	43	List of procedures
DFHA60H	DFHA6002	43	Sequence of operations
DFHA60I	DFHA6002	45	Start up CICS
DFHA60J	DFHA6002	46	Chart A-3: CICS initialization with local DL/I
DFHA60K	DFHA6002	48	Normal CICS system monitoring using master terminal commands
A4CHAR	DFHA6002	51	Chart A-4: Shut down CICS
DFHA60L	DFHA6002	54	Chart B-1: Action on user problem
DFHA60M	DFHA6002	57	Chart B-2: IMS database backout failure
DFHA60N	DFHA6002	60	Chart B-3: IMS database medium failure
DFHA60O	DFHA6002	63	Chart B-4: IRLM connection failure
DFHA60P	DFHA6002	66	Chart B-5: IRLM communications failure
DFHA60Q	DFHA6002	69	Chart B-6: Machine check, power failure, MVS loop, or abend
DFHA60R	DFHA6002	71	Errors that do not cause CICS to abend
DFHA60S	DFHA6002	73	Abend errors
DFHA60T	DFHA6002	74	Perform emergency restart
DFHA60U	DFHA6002	76	Chart C-1: Log errors
DFHO103	DFHA6003	79	Chapter 5, Running a batch region for DL/I shared databases 1
DFHA60W	DFHA6003	81	Coding an EXEC PGM=DFHDRP statement 80
DFHA60X	DFHA6004	83	Chapter 6, Operating CICS in an XRF environment 1
DFHA60Y	DFHA6004	83	Overview of the CICS extended recovery facility (XRF) 83
DFHA60Z	DFHA6004	85	Functions of the CEMT and CEBT transactions for XRF 83
DFHA610	DFHA6004	88	Starting up and synchronizing an XRF system 83
DFHA611	DFHA6SXR	90	Shutting down CICS XRF regions 83, 129
DFHA612	DFHA6SXR	94	Takeover 83

DFHA613	DFHA6SXR	95	What events can cause a takeover? 94
DFHA614	DFHA6SXR	97	What happens during a takeover 110
DFHA615	DFHA6SXR	98	Action on takeover delay—multi-MVS/ESA SP 4.1 environment
DFHA616	DFHA6SXR	101	Action on takeover delay—single MVS/ESA SP 4.1 image
DFHA617	DFHA6SXR	101	Action on takeover delay—MVS earlier than MVS/ESA SP 4.1
CTAKOVR	DFHA6SXR	103	Completing the takeover 98
DFHA619	DFHA6SXR	103	Post-takeover processing
DFHA61A	DFHA6SXR	103	Takeover after a command 83, 95
DFHA61B	DFHA6SXR	106	Takeover after abnormal sign-off 83, 95
DFHA61C	DFHA6SXR	108	Takeover after missing surveillance signal 83, 95
DFHA61D	DFHA6SXR	111	Takeover after MVS failure (MVS/ESA SP 4.1) 83, 95, 108
DFHA61E	DFHA6SXR	111	What happens if the alternate CICS fails 83
DFHA61F	DFHA6SXR	112	Operating an XRF overseer 83
DFHA61G	DFHA6SXR	112	MVS MODIFY commands to control the overseer
OVSRRPRE	DFHA6SXR	115	CICS overseer modules
DFHA61I	DFHA6SXR	116	Using a DFHOSD data set to tell the overseer what to monitor
OVSRRJOB	DFHA6SXR	118	Job control statements to run the CICS overseer
DFHA601	DFHA6SXR	120	A run-time example
DFHA61N	DFHA6006	121	Chapter 7, Operating CICS from a console device 1
DFHA6CM	DFHA6006	122	Entering commands from a console device
JCLCONS	DFHA6006	122	Using JCL to initiate CICS commands 121
DFHA61P	DFHA6006	124	Suppressing information-only messages
DFHA6Q	DFHA6006	124	Replying to messages
DFHA6QL	DFHA6006	125	Replying to messages from transactions started at console devices 122
DFHA6SD	DFHA6SHT	127	Chapter 8, How to shut down CICS xx, 1, 3, 126
SEEIM	DFHA6SHT	128	What you should see
PART2	DFHA6PT2	131	Part 2, The CICS utility programs xi, 13
DFHO201	DFHA6007	133	Chapter 9, Journal utility programs
DFHA61Y	DFHA6007	133	The DFHJUP utility program 131
JUPOPT	DFHA6007	137	OPTION statement 155
DFHA61Z	DFHA6007	141	Examples of using DFHJUP 134
AJARCH	DFHA6007		

		146	The automatic journal archiving utility 45, 131, 133
DFHA621	DFHA6007	148	Symbolic parameter substitution 146
DFHO202	DFHA6008	151	Chapter 10, Statistics utility program, DFHSTUP 131
DFHA623	DFHA6008	153	Job to run the DFHSTUP program 152
STUPRM	DFHA6008	156	Control parameters of the DFHSTUP program 156
DFHO203	DFHA6009	163	Chapter 11, Trace utility print programs 131
TUPARM	DFHA6009	164	The trace selection parameters 164, 182, 191
TRATACH	DFHA6009	167	Identifying trace entries from their transaction-attach entries 166, 166
DFHA642	DFHA6009	169	Using IPCS to print trace records written to GTF
DFHO204	DFHA6010	173	Chapter 12, Dump utility programs
CICSDUP	DFHA6010	173	The CICS transaction dump utility program, DFHDU410 131, 173
PRDMPCS	DFHA6010	179	Using IPCS to format and analyze CICS dumps 131, 173
CDOVERO	DFHA6010	180	The SDUMP options needed to support the CICS dump exit 173, 179
DFHA62B	DFHA6010	180	Specifying DFHIPCSP CICS exit control data needed 179, 194
DFHA62C	DFHA6010	180	Renaming dump exit routines for releases before CICS/ESA 4.1 180, 181
DFHA62D	DFHA6010	181	Making available CICS dump exit routines needed 180
D5090D	DFHA6010	182	Selecting parts of the CICS internal trace table 179
DFHA62E	DFHA6010	183	The CICS dump exit parameters 182
IPCSXMP	DFHA6010	184	Examples of the use of level numbers 182
DFHA641	DFHA6010	191	Trace entry selection
DUMPSUM	DFHA6010	192	The dump summary and error index 184
DFHA62H	DFHA6011	197	Chapter 13, Monitoring utility programs
DFHA62I	DFHA6011	197	The monitoring dictionary utility program, DFHMNDUP 131, 213
DFHA62J	DFHA6011	197	The performance dictionary record 203
DFHA62K	DFHA6011	202	The sample monitoring data print program, DFH\$MOLS 131
DFHA62L	DFHA6011	203	Monitoring dictionary records 213, 214
DFHA62M	DFHA6011	205	Control statements of the DFH\$MOLS program 201, 203, 205
DFHO206	DFHA6012	215	Chapter 14, System definition file utility program, DFHCSDUP 131
DFHA62O	DFHA6012	215	Sharing the CSD between CICS/ESA 4.1 and earlier releases

			217, 220
INVBTC	DFHA6012	217	Invoking DFHCSDUP as a batch program 215
INVUSER	DFHA6012	219	Invoking the DFHCSDUP program from a user program 215
DFHA62R	DFHA6012	221	Commands for the DFHCSDUP program
DFHA62S	DFHA6012	221	Rules for the syntax and preparation of commands
DFHABC	DFHA4ADD	223	ADD a group to a list
DFHABD	DFHA4ALT	224	ALTER a resource definition
DFHA4YB	DFHA4APP	225	APPEND a list to another list
DFHABE	DFHA4COP	227	COPY a resource definition
GENCOP	DFHA4COP	228	Generic naming in the COPY command 227, 227
DFHA4ZI	DFHA4DEF	229	DEFINE a resource definition
DELL	DFHA4DEL	231	DELETE a resource definition, group, or list
DFHA42E	DFHA4EXT	233	EXTRACT resource definition data
DFHABF	DFHA4INI	235	INITIALIZE a new CSD with the required CICS resource definitions
SPLAT	DFHA4MIG	238	MIGRATE macro-defined resource definitions from tables to the CSD
REMOVE	DFHA4REM	241	REMOVE a group from a list 231
DFHABH	DFHA4SER	242	SERVICE a CSD
DFHA4AB	DFHA4UPG	243	UPGRADE a CSD
DFHABI	DFHA4VER	244	VERIFY a CSD
DFHO207	DFHA6013	245	Chapter 15, Identify macro-level programs utility program, DFHMSCAN 131
DFHA62W	DFHA6013	246	Summary report 246
DFHA62X	DFHA6013	247	Detailed report 246
DFHA62Y	DFHA6013	248	Limitations of the DFHMSCAN program
DFHO208	DFHA6014	249	Chapter 16, Sample resource manager in-doubt resolution program, DFH\$INDB 131
DFHO209	DFHA6015	253	Chapter 17, Signon table to RACF migration utility program, DFH\$NMIG 131
DFHA631	DFHA6015	254	Sample job stream to run the DFH\$NMIG program 253
DFHA632	DFHA6015	254	Sample batch job to execute the CLIST 253
DFHO210	DFHA6016	255	Chapter 18, Stagger end-of-day time sample utility program, DFH\$STED 131
DFHMEU	DFHLA613 SCRIPT	257	Chapter 19, Message editing utility 131
MEUDS	DFHE4ICD	257	Utility data sets 257
MECL	DFHE4ICD	258	Defining the utility data set index

			257, 257
EXEC	DFHE4ICD	259	Using the message editing utility
MEUST	DFHE4ICD	259	1. Starting the message editing utility 259
DFLTS	DFHE4ICD	260	2. Specifying default values for the message editing utility 259
MEUSL	DFHE4ICD	262	Selecting languages for message translation
MEUACT	DFHE4ICD	264	3. Performing actions on message data sets 259
EDITSEL	DFHE4ICD	266	Selecting message sets to be edited
MEUSTEP	DFHE4ICD	268	4. Add the new message load modules to STEPLIB 259
PTF	DFHE4ICD	269	5. Applying PTFs to the message editing utility 259
PTFLOG	DFHE4ICD	270	PTF update log sample output 270
ERULES	DFHE4ICD	271	Rules for editing and translating 267
HELP	DFHE4ICD	274	Getting help
IWUP	DFHLA613 SCRIPT	277	Chapter 20, In-doubt window resolution utility, DFH\$IWUP 131
NORESY	DFHE4ICX	278	Limitations of the DFH\$IWUP utility program
DFHA6BP	DFHA6017	287	Chapter 21, Local catalog storage program, DFHSMUTL 131
DFHO101	DFHA6001	291	Appendix A, Operating procedures xi, xx, 1, 42
DFHA604	DFHA6001	291	Departmental and job titles 291
DFHA6XN	DFHLA613 SCRIPT	305	Appendix B, CICS messages that cannot be edited xi
DFHA634	DFHLA613 SCRIPT	309	Index

Index Entries

<u>id</u>	<u>File</u>	<u>Page</u>	<u>References</u>
IAUTJA	DFHA6SET	i	(1) automatic journal archiving
IJNLDS	DFHA6SET	i	(1) journal data sets
ISVC	DFHA6SET	i	(1) SVC 23, 79
ISVC3	DFHA6SET	i	(1) DFHCSVC, the CICS type 3 SVC 23, 79
JDS	DFHA6SET	i	(1) journal data sets 45
DBRCI	DFHA6SET	i	(1) DBRC (database recovery control) 31, 61
JNLARCH	DFHA6SET	i	(1) journal automatic archiving 52, 58, 61, 146
IIRC	DFHA6SET	i	(1) IRC (interregion communication) 26
IMRO	DFHA6SET	i	(1) MRO (multiregion operation) 26

ICLT	DFHA6SET	i	(1) CLT (command list table) 103, 106, 106
ICLT2	DFHA6SET	i	(1) command list table (CLT) 103, 106, 106
IRACF	DFHA6SET	i	(1) RACF (resource access control facility) 119, 253
IXRF	DFHA6SET	i	(1) XRF (extended recovery facility) 37, 83, 85, 85, 85, 88, 88, 88, 89, 91, 92, 94, 95, 97, 97, 98, 103, 103, 106, 108, 111, 112, 112, 114, 115, 115, 115, 115, 116
ITRC	DFHA6SET	i	(1) trace 163
ICSD	DFHA6SET	i	(1) CSD (CICS system definition file) 215, 215
ISTRTSK	DFHA6STR	19	(1) started task, CICS as a
IMSDSH	DFHA6002	29	(1) IMS data sharing 29, 34, 34, 37, 43
IIRLM	DFHA6002	37	(1) IMS resource lock manager (IRLM) 37, 53, 56, 56, 56, 56, 56, 64, 64, 64, 65, 65, 65, 67, 67, 67, 68
FLAWS	DFHA6002	42	(1) IMS operating procedures and flowcharts 42, 42, 43, 45, 70, 70
IDLIOP	DFHA6003	79	(1) DL/I shared database operation 79, 80
IDRP	DFHA6003	79	(1) DFHDRP program, CICS-DL/I batch-region controller 81
IGTF	DFHA6004	87	(1) generalized trace facility (GTF) 87, 169, 169, 170, 170, 171
ITAK	DFHA6SXR	94	(1) XRF (extended recovery facility) (2) takeover 95, 97, 97, 98, 103, 103, 103, 106, 108, 111
DFHJUP	DFHA6007	133	(1) journal utility program, DFHJUP 133, 133, 133, 134, 135, 136
JUPCON	DFHA6007	136	(1) journal utility program, DFHJUP (2) control statements 136, 137, 141, 141, 141
DFHSTUP	DFHA6008	151	(1) DFHSTUP statistics utility program 151, 153, 156
DFHTUP	DFHA6009	163	(1) DFHTU410 trace utility program 164
TUPCON	DFHA6009	164	(1) DFHTU410 trace utility program (2) selection parameters 164, 164, 165, 165, 165, 165, 165, 165, 165, 166, 166, 166, 166, 166, 167
IIPCS	DFHA6009	169	(1) interactive problem control system (IPCS) 169, 179, 180, 180, 181, 192, 192, 192, 192, 192
DFHDUP	DFHA6010	173	(1) DFHDU410 transaction dump utility program 173, 174, 177
A62DMP	DFHA6010	192	(1) interactive problem control system (IPCS) (2) dump summary 192
DFH\$MOL	DFHA6011	197	(1) DFH\$MOLS sample print program 202, 202, 204, 205
MOLCON	DFHA6011	205	(1) DFH\$MOLS sample print program (2) control statements 206, 206, 206, 208, 209, 210, 210, 211
ICSDUP	DFHA6012		

		215	(1) DFHCSDUP system definition utility program 217, 219, 222
CSDUPC	DFHA6012	222	(1) DFHCSDUP system definition utility program 222, 223, 224, 226, 228, 230, 232, 234, 235, 237, 240, 241, 242, 243
CSDEXT	DFHLA613 SCRIPT	232	(1) EXTRACT command, DFHCSDUP utility program
CSDREM	DFHLA613 SCRIPT	240	(1) REMOVE command, DFHCSDUP utility program
CSDUPG	DFHLA613 SCRIPT	242	(1) UPGRADE command, DFHCSDUP utility program
MACLVL	DFHA6013	245	(1) macro-level programs, identifying
DOUBT	DFHA6014	249	(1) in-doubt resolution program
SNMIG	DFHA6015	253	(1) SNT to RACF migration utility program
ISNMIGU	DFHA6015	253	(1) DFHSNMIG utility program, to migrate from SNT to RACF database 254, 254
SEOT	DFHA6016	255	(1) staggering the end-of-day time
ISEOTU	DFHA6016	255	(1) DFH\$STED, utility to stagger the end-of-day time
DFHMEUI	DFHLA613 SCRIPT	257	(1) message editing utility 257, 257, 258, 259, 260, 262, 264, 266, 267, 269, 271, 274
RUNMEU	DFHE4ICD	259	(1) message editing utility (2) using 259, 260, 262, 264, 265, 266, 267, 268, 268, 268, 269, 270, 270, 271, 274
IFORMS	DFHA6001	291	(1) forms and tables for operating procedures 291, 291, 293, 295, 297
CHARTS	DFHA6001	299	(1) flowcharts for error procedures 300, 302
ERRPROC	DFHA6001	299	(1) error procedures 300, 302

List Items

<u>id</u>	<u>File</u>	<u>Page</u>	<u>References</u>
MEUCPY	DFHE4ICD	259	

Footnotes

<u>id</u>	<u>File</u>	<u>Page</u>	<u>References</u>
CPC	DFHA6005	25	1 25
PRSMDFN	DFHA6004	84	2 84
SYSPFN	DFHA6SXR	100	4 100

NN	DFHA6SET	i	8, 8, 39, 39, 119, 119, 119, 119, 135, 135, 135, 135, 141, 142, 142, 146, 153, 153, 153, 153, 154, 154, 156, 156, 156, 174, 174, 175, 175, 177, 177, 217, 217, 217, 217, 217, 220, 220, 220, 220, 246, 246, 248, 248
MM	DFHA6SET	i	xix, xix, xx, xx, 151, 151, 151, 151, 151, 152, 154, 154, 156, 156, 156, 156, 159, 159, 160, 160, 160, 160, 167, 167, 169, 169, 170, 170, 171, 171, 171, 171, 171, 171, 171, 171, 171, 171, 172, 172, 172, 178, 178, 178, 178, 178, 178, 179, 179, 179, 179, 194, 194, 194, 194, 194, 194, 194, 194, 195, 195, 198, 198, 198, 198, 198, 199, 205, 205, 209, 209, 209, 209, 223, 223, 224, 224, 224, 224, 225, 225, 238, 238, 249, 249, 251, 251, 255, 255, 261, 261
M1	DFHA6SET	i	122, 122, 131, 131, 135, 135, 135, 135, 142, 142, 157, 157, 157, 161, 161, 180, 180, 185, 191, 194, 194, 199, 199, 213, 213, 234, 234, 240, 240, 255, 255, 260, 260, 262, 262, 262, 262, 262, 262, 262, 262, 262, 285, 289

Spots

<u>id</u>	<u>File</u>	<u>Page</u>	<u>References</u>
BARS	DFHA6SOC	xix	(no text)
PRT1ST	DFHA6PT1	1	(no text)
STARTS	DFHA6STR	18	(no text)
FLOWIM	DFHA6002	43	(no text)
XA1	DFHA6002	43	(no text)
XA2	DFHA6002	45	(no text)
XA3	DFHA6002	46	(no text)
XX1	DFHA6002	48	(no text)
XX2	DFHA6002	51	(no text)
A4CHAR	DFHA6002	51	chart A-4
XB1	DFHA6002	54	(no text)
XB2	DFHA6002	57	(no text)
DLISTOP	DFHA6002	58	the second note for Chart B-2
XB3	DFHA6002	60	(no text)
XB4	DFHA6002	63	(no text)
XB5	DFHA6002	66	(no text)
XB6	DFHA6002	69	(no text)
XB7	DFHA6002	71	(no text)

			43
XB8	DFHA6002	73	(no text) 43
XXC	DFHA6002	74	(no text) 43
N1CHAR	DFHA6002	74	(no text)
XC1	DFHA6002	76	(no text) 43
MVSFAIL	DFHA6SXR	99	(no text) 97
STEP3	DFHA6SXR	105	(no text) 107, 108, 110, 110
STEP4	DFHA6SXR	105	(no text) 109
TKAUTO	DFHA6SXR	109	(no text)
REST	DFHA6SXR	114	(no text) 120
RUN	DFHA6SXR	120	(no text)
PRT1END	DFHLA613 SCRIPT	129	(no text) xi
PRT2ST	DFHA6PT2	131	(no text) xi
JUPSTG	DFHA6007	134	(no text) 135
RTNCDE	DFHA6007	147	(no text)
STNOTE	DFHA6008	154	(no text) 204
DFHA4AA	DFHA6010	180	(no text) 194
DEFSCOP	DFHA6010	184	(no text)
PDXIDS	DFHA6010	185	(no text) 182, 182
DFHA6K7	DFHA6010	185	(no text) 185
MONAMP	DFHA6011	201	(no text) 154
CTLDES	DFHA6011	206	(no text) 205
MIGREG	DFHA6012	217	(no text)
CBDOUT	DFHA6012	218	(no text)
DFHA4BQ	DFHA4ADD	223	(no text)
DFHA4BS	DFHA4ALT	224	(no text)
DFHA4BU	DFHA4APP	225	(no text)
DFHA4BX	DFHA4COP	227	(no text)
CSDDEF	DFHA4DEF	229	(no text)
DFHA4BZ	DFHA4DEF	229	(no text)
DFHA4CB	DFHA4DEL	231	(no text)
DFHA4CE	DFHA4EXT	233	(no text)
DFHA4CF	DFHA4INI	235	(no text)

DFHA4CH	DFHA4LIS	236	(no text)
DFHA4CJ	DFHA4MIG	238	(no text)
DFHA4CM	DFHA4REM	241	(no text)
DFHA4CO	DFHA4SER	242	(no text)
DFHA4CQ	DFHA4UPG	243	(no text)
DFHA4CS	DFHA4VER	244	(no text)
MEUU	DFHE4ICD	259	(no text)
PRT2END	DFHLA613 SCRIPT	285	(no text)
A1	DFHA6001	300	(no text)
A2	DFHA6001	302	(no text)

Tables

<u>id</u>	<u>File</u>	<u>Page</u>	<u>References</u>
LIBNAMS	DFHA6SOC	xxi	1
AUTHOR	DFHA6002	33	xx
ACCESSD	DFHA6002	33	2
TKACT	DFHA6004	86	32
TKALT	DFHA6004	86	3
TAKOVRS	DFHA6SXR	97	4
OSDRECS	DFHA6SXR	116	85, 85
TABCON	DFHA6007	136	5
TABOPT	DFHA6007	137	85
TABEND	DFHA6007	141	6
TABCOM	DFHA6007	141	96, 97
OPVAL	DFHA6007	145	7
STUPCPS	DFHA6008	157	116, 119
CDOM	DFHA6009	167	8
MN2SMF	DFHA6011	198	9
SAMPDDS	DFHA6012	218	10
MELANG	DFHE4ICD	263	11

Processing Options

Runtime values:

Document fileid	DFHLA613 SCRIPT
Document type	USERDOC
Document style	IBMXAGD
Profile	EDFPRF40
Service Level	0032
SCRIPT/VS Release	4.0.0
Date	00.01.19
Time	11:08:59
Device	PSA
Number of Passes	3
Index	YES
SYSVAR B	SPINE
SYSVAR G	INLINE
SYSVAR P	PS
SYSVAR R	DFHLA613
SYSVAR W	DFHLA613
SYSVAR X	YES

Formatting values used:

Annotation	NO
Cross reference listing	YES
Cross reference head prefix only	NO
Dialog	LABEL
Duplex	YES
DVCF conditions file	(none)
DVCF value 1	(none)
DVCF value 2	(none)
DVCF value 3	(none)
DVCF value 4	(none)
DVCF value 5	(none)
DVCF value 6	(none)
DVCF value 7	(none)
DVCF value 8	(none)
DVCF value 9	(none)
Explode	NO
Figure list on new page	YES
Figure/table number separation	YES
Folio-by-chapter	NO
Head 0 body text	Part
Head 1 body text	Chapter
Head 1 appendix text	Appendix
Hyphenation	NO
Justification	NO
Language	ENGL
Keyboard	395
Layout	OFF
Leader dots	YES
Master index	(none)
Partial TOC (maximum level)	2
Partial TOC (new page after)	INLINE
Print example id's	NO
Print cross reference page numbers	YES
Process value	PS
Punctuation move characters	,
Read cross-reference file	DFHLA613
Running heading/footer rule	NONE
Show index entries	NO
Table of Contents (maximum level)	2
Table list on new page	YES
Title page (draft) alignment	RIGHT
Write cross-reference file	DFHLA613

Named Files

<u>Fileid</u>	<u>Page</u>	<u>Data Set Name</u>
RCF118	4	RCF118 EPSBIN

Imbed Trace

Page 0	DFHA6SET
Page i	DFHL99E2
Page vi	DFHL99SP
Page vii	DFHL99TM
Page vii	DFHA6TMK
Page vii	DFHA6PRE
Page xi	DFHQ99CU
Page xi	DFHL99L2
Page xiii	DFHA6BIB
Page xvii	DFHA6SOC
Page xxii	DFHA6PT1
Page 1	DFHA6OVW
Page 16	DFHA6STR
Page 21	DFHA6CPM
Page 24	DFHA6005
Page 27	DFHA6002
Page 78	DFHA6003
Page 82	DFHA6004
Page 90	DFHA6SXR
Page 120	DFHA6006
Page 126	DFHA6SHT
Page 129	DFHA6PT2
Page 131	DFHA6007
Page 149	DFHA6008
Page 161	DFHA6009
Page 172	DFHA6010
Page 195	DFHA6011
Page 214	DFHA6012
Page 222	DFHA4ADD
Page 223	DFHA4ALT
Page 224	DFHA4APP
Page 225	DFHA4104
Page 226	DFHA4COP
Page 228	DFHA4DEF
Page 230	DFHA4DEL
Page 232	DFHA4EXT
Page 234	DFHA4INI
Page 235	DFHA4LIS
Page 237	DFHA4MIG
Page 240	DFHA4REM
Page 241	DFHA4SER
Page 242	DFHA4UPG
Page 243	DFHA4VER
Page 244	DFHA6013
Page 248	DFHA6014
Page 252	DFHA6015
Page 254	DFHA6016
Page 257	DFHE4ICD
Page 277	DFHE4ICE
Page 278	DFHE4ICX
Page 279	DFHE4ICY
Page 280	DFHE4ICZ
Page 280	DFHE4IDA
Page 284	DFHE4IDB
Page 285	DFHE4IDC
Page 285	DFHA6017
Page 290	DFHA6001
Page 305	DFHA6MXN
Page 315	RCFFAMILY

IMBED OR APPEND FILE NOT FOUND.

'EDF#INIT' LINE 1850: .im DSMREFS

'EDF#INIT' WAS IMBEDDED AT LINE 1850 OF 'EDF#INIT'

'EDF#INIT' WAS IMBEDDED AT LINE 360 OF 'EDF#MAIN'

'EDF#MAIN' WAS IMBEDDED AT LINE 177 OF 'EDFPRF40'

'EDFPRF40' WAS IMBEDDED AT LINE 0 OF 'DFHLA613'

IMBED OR APPEND FILE NOT FOUND.

'EDFTOC' LINE 340: .im DSMTOC

'EDFTOC' WAS IMBEDDED AT LINE 340 OF 'EDFTOC'

'EDFTOC' WAS IMBEDDED AT LINE 11 OF 'DFHLA613'

STARTING PASS 2 OF 3.

STARTING PASS 3 OF 3.