

CICS<sup>®</sup> Transaction Server for OS/390<sup>®</sup>



# CICS Operations and Utilities Guide

*Release 3*



CICS<sup>®</sup> Transaction Server for OS/390<sup>®</sup>



# CICS Operations and Utilities Guide

*Release 3*

**Note!**

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

**Fourth edition (November 2000)**

This edition applies to Release 3 of CICS Transaction Server for OS/390, program number 5655-147, and to all subsequent versions, releases, and modifications until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

This edition replaces and makes obsolete the previous edition, SC33-1685-33. Changes since that edition are indicated by a # sign to the left of a change. Any vertical lines in the left margin indicate a change made between Version 1 Release 2 and Version 1 Release 3 of CICS Transaction Server for OS/390.

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, Information Development,  
Mail Point 95, 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 1989, 2000. All rights reserved.

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

---

# Contents

<b>Notices</b> . . . . .	<b>vii</b>
Trademarks . . . . .	viii

<b>Preface</b> . . . . .	<b>ix</b>
What this book is about . . . . .	ix
Who should read this book? . . . . .	ix

<b>Bibliography</b> . . . . .	<b>xi</b>
CICS Transaction Server for OS/390 . . . . .	xi
CICS books for CICS Transaction Server for OS/390 . . . . .	xi
CICSplex SM books for CICS Transaction Server for OS/390 . . . . .	xi
Other CICS books . . . . .	xii

<b>Summary of changes</b> . . . . .	<b>xiii</b>
Changes for the CICS Transaction Server for OS/390 Release 3 edition . . . . .	xiii
Changes for the CICS Transaction Server for OS/390 Release 2 edition . . . . .	xiii
Changes for the CICS Transaction Server for OS/390 release 1 . . . . .	xiv

---

## Part 1. Operating CICS® regions . . . 1

### Chapter 1. Overview of CICS operations 3

Starting up CICS . . . . .	3
The types of CICS startup . . . . .	3
CICS actions on an initial start . . . . .	4
CICS actions on a cold start . . . . .	5
CICS actions on a warm start . . . . .	5
CICS actions on an emergency restart . . . . .	8
CICS startup and the VTAM® session . . . . .	8
End of CICS startup . . . . .	10
Controlling CICS operation . . . . .	10
Controlling CICS with CICSplex SM . . . . .	10
CICS-supplied transactions . . . . .	11
CICS-supplied utility programs . . . . .	12
Shutting down CICS . . . . .	12
CICS XRF systems . . . . .	12
Normal shutdown (PERFORM SHUTDOWN) . . . . .	12
Immediate shutdown (PERFORM SHUTDOWN IMMEDIATE) . . . . .	14
Uncontrolled shutdown . . . . .	15

### Chapter 2. Starting up CICS regions . . 17

Starting CICS . . . . .	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 . . . . .	20
Rules for coding parameters at the console . . . . .	21

Entering corrections to parameters at the console . . . . .	21
System console messages for CICS startup . . . . .	21

### 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. Operating CICS from a console device . . . . . 29

Entering commands from a console device . . . . .	29
Entering commands from TSO . . . . .	30
Using JCL to initiate CICS commands . . . . .	30
Console device messages . . . . .	30
Console message-formatting . . . . .	31
Suppressing information-only messages . . . . .	32
Replying to messages . . . . .	32
Replying to messages from transactions started at console devices . . . . .	32
Suppressing and rerouting messages . . . . .	33
Sample console messages for CICS startup . . . . .	33
Sample console messages for CICS shutdown . . . . .	33

### Chapter 5. How to shut down CICS . . . 35

Shutting down CICS normally . . . . .	35
Shutting down CICS immediately . . . . .	36
Shutting down XRF CICS regions . . . . .	36

---

## Part 2. The CICS utility programs 39

### Chapter 6. Log stream sizing migration utility, DFHLSCU . . . . . 41

Output from DFHLSCU . . . . .	41
Job control statements to run the DFHLSCU program . . . . .	43
SYSIN control statements for the DFHLSCU utility . . . . .	44
Format of the SYSIN control statements . . . . .	44
Considerations when using DFHLSCU . . . . .	45
DFHLSCU return codes . . . . .	46
Output from the DFHLSCU utility . . . . .	47

### Chapter 7. Using batch jobs to read log streams . . . . . 53

SUBSYS=(LOGR,DFHLGCNV,...) keyword . . . . .	53
Using DFHJUP to read log streams . . . . .	57
The DD statements . . . . .	58
Utility control statements . . . . .	59
DFHJUP return codes . . . . .	65

Managing the size of log streams . . . . .	65
Log data accessible to DFHJUP . . . . .	66
Diagnostic information in DFHJUP output . . . . .	67
Examples of using DFHJUP . . . . .	68

**Chapter 8. Statistics utility program (DFHSTUP) . . . . . 77**

The statistics recording status . . . . .	77
Restriction in the use of some output devices . . . . .	78
Job to run the DFHSTUP program . . . . .	78
Control parameters of the DFHSTUP program . . . . .	83

**Chapter 9. Recovery statistics program (DFH\$STER) . . . . . 91**

Recovery Manager domain recovery messages . . . . .	91
Statistics sample program, DFH\$STER . . . . .	92
Installation . . . . .	92

**Chapter 10. Trace utility print program (DFHTU530) . . . . . 93**

The CICS trace utility program, DFHTU530 . . . . .	93
The trace selection parameters . . . . .	94
Using IPCS to print trace records written to GTF . . . . .	99
The GTFTRACE subcommand of IPCS and associated parameters . . . . .	100

**Chapter 11. Dump utility program (DFHDU530) . . . . . 103**

The CICS transaction dump utility program, DFHDU530 . . . . .	103
Selecting transaction dump output for the DFHDU530 program . . . . .	103
Job control statements to run the DFHDU530 program . . . . .	106
Using IPCS to format and analyze CICS dumps . . . . .	108
Preparing to use IPCS to format CICS SDUMPs . . . . .	109
Selecting parts of the CICS internal trace table . . . . .	111
Using CICS-supplied dump exit routines to format CICS SDUMPs . . . . .	111
The dump summary and error index . . . . .	115
Sample jobs to process a CICS SDUMP using the CICS dump exit . . . . .	116

**Chapter 12. Monitoring utility programs (DFHMNDUP and DFH\$MOLS) . . . . . 119**

The monitoring dictionary utility program, DFHMNDUP . . . . .	119
The performance dictionary record . . . . .	119
Sample job illustrating the use of DFHMNDUP . . . . .	122
The sample monitoring data print program, DFH\$MOLS . . . . .	124
Overview of the DFH\$MOLS program . . . . .	124
Sample job stream for the DFH\$MOLS programs . . . . .	125
Control statements of the DFH\$MOLS program . . . . .	127
Rules for coding control statements . . . . .	129
Abend codes and error messages for the DFH\$MOLS program . . . . .	136

**Chapter 13. System definition file utility program (DFHCSDUP) . . . . . 139**

Sharing the CSD between CICS Transaction Server for OS/390 Release 3 and earlier releases . . . . .	140
Input to the DFHCSDUP program . . . . .	140
Output from the DFHCSDUP program . . . . .	140
Invoking DFHCSDUP as a batch program . . . . .	141
Invoking the DFHCSDUP program from a user program . . . . .	144
Entry parameters for the DFHCSDUP program . . . . .	145
Responsibilities of the user program . . . . .	146

**Chapter 14. Commands for the DFHCSDUP program . . . . . 147**

Rules for the syntax and preparation of commands . . . . .	147
Command processing following internal error detection . . . . .	148
ADD . . . . .	149
ALTER . . . . .	150
APPEND . . . . .	152
COPY . . . . .	154
DEFINE . . . . .	156
DELETE . . . . .	158
EXTRACT . . . . .	160
INITIALIZE . . . . .	162
LIST . . . . .	163
MIGRATE . . . . .	165
PROCESS . . . . .	169
REMOVE . . . . .	170
SCAN . . . . .	171
SERVICE . . . . .	173
UPGRADE . . . . .	174
VERIFY . . . . .	175

**Chapter 15. Load module scanner (DFHEISUP) . . . . . 177**

Overview . . . . .	177
Creating a summary report . . . . .	178
Creating a detailed report . . . . .	179
Filter input format definition . . . . .	183
Activating trace . . . . .	184

**Chapter 16. Batch-enabling sample programs for RLS access-mode data sets . . . . . 187**

Switching from RLS to non-RLS access mode . . . . .	187
Preparing data sets for batch operations . . . . .	187
Installing the sample programs . . . . .	190
Preparing input for the sample programs . . . . .	190

**Chapter 17. Identify macro-level programs utility program (DFHMSCAN) . . . . . 191**

Overview . . . . .	191
Running DFHMSCAN . . . . .	191
How DFHMSCAN works . . . . .	192
Using DFHMSCAN . . . . .	192
Summary report . . . . .	192

Detailed report . . . . .	193
Limitations of the DFHMSCAN program . . . . .	194

**Chapter 18. Sign-on table to RACF migration utility program (DFHSNMIG) 195**

Migrating operator characteristics—CICS SNT to RACF database . . . . .	195
Sample job stream to run the DFHSNMIG program	196
Sample batch job to execute the CLIST . . . . .	196

**Chapter 19. Stagger end-of-day time sample utility program (DFH\$STED). 197**

Example . . . . .	198
-------------------	-----

**Chapter 20. Message editing utility (DFHMEU). 199**

Requirements . . . . .	199
Installing the message editing utility . . . . .	199
Utility data sets . . . . .	199
Defining the utility data set index . . . . .	200
Using the message editing utility . . . . .	200
Restriction on running the message editing utility . . . . .	201
1. Starting the message editing utility . . . . .	201
2. Specifying default values for the message editing utility . . . . .	201
3. Performing actions on message data sets . . . . .	205
4. Add the new message load modules to STEPLIB . . . . .	209
5. Applying PTFs to the message editing utility	210
Rules for editing and translating . . . . .	212
Getting help . . . . .	215

**Chapter 21. Shutdown assist program (DFHCESD) 217**

Actions of the default program . . . . .	217
The sample programs . . . . .	218

**Chapter 22. Recovery manager utility program (DFHRMUTL) 223**

Overview of DFHRMUTL . . . . .	223
Input to DFHRMUTL . . . . .	223
Output from DFHRMUTL . . . . .	223
JCL requirements . . . . .	224

DD statements . . . . .	224
Specifying parameters . . . . .	224
Return codes . . . . .	226
Examples of using DFHRMUTL . . . . .	226
Setting an initial start without operator intervention . . . . .	226
Examining the override record . . . . .	227
Resetting a warm or emergency start . . . . .	227
Improving the performance of a cold start. . . . .	228

**Chapter 23. BMS macro generation utility program (DFHBMSUP). 229**

Overview. . . . .	229
Input . . . . .	229
Output . . . . .	229
DD statements . . . . .	229
Return codes . . . . .	230
Example of using DFHBMSUP . . . . .	230
Example of DFHBMSUP output . . . . .	230

**Chapter 24. The Transaction Affinities Utility. 233**

**Chapter 25. Offsite Automatic Reply program (DFH\$OFAR). 235**

Overview of DFH\$OFAR . . . . .	235
Control file definition . . . . .	235
Netview configuration . . . . .	236
Control file typical settings . . . . .	236
Program failures . . . . .	236

**Chapter 26. Local catalog storage program, DFHSMUTL 237**

Input . . . . .	237
Output . . . . .	237
Messages . . . . .	237
Job control statements to run the DFHSMUTL program . . . . .	239

**Part 3. Appendixes 241**

**Index 243**

**Sending your comments to IBM 249**



---

## Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation  
Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106, Japan

**The following paragraph does not apply in the United Kingdom or any other country where such provisions are inconsistent with local law:**

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore this statement may not apply to you.

This publication could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM United Kingdom Laboratories, MP151, 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.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Programming License Agreement, or any equivalent agreement between us.

---

## Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

AD/Cycle	CICSplex	IMS/ESA
AT	DB2	MVS/ESA
BookManager	DFSMS	OS/390
C/370	DFSMS/MVS	PR/SM
CICS	DFSORT	RACF
CICS/ESA	IBM	S/370
CICS/MVS	IBMLink	SP
CICS/VSE	IMS	VTAM

Other company, product, and service names may be trademarks or service marks of others.

---

## Preface

---

### What this book is about

This book is intended to help you operate CICS regions in an MVS environment. It contains guidance about operating CICS regions in an MVS environment, particularly when using multiregion operation (MRO). It also contains guidance about how to use the CICS batch utility programs.

#### **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 SM Concepts and Planning*, GC33-0786.
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. For further information, see the *CICS Transaction Affinities Utility Guide*.

### Who should read this book?

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 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 “CICS books for CICS Transaction Server for OS/390” on page xi.)

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** The CICS element of the IBM CICS Transaction Server for OS/390.

**MVS** The MVS operating system.

**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 and is assumed to be assigned the EBCDIC code point X'5B'. In some countries a different currency symbol, for example the pound symbol (£), or the yen symbol (¥), is assigned the same EBCDIC code point. In these countries, the appropriate currency symbol should be used instead of the dollar symbol.

**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 usually in step. However, due to the time required to print and distribute hardcopy books, the BookManager version is more likely to have had last-minute changes made to it before publication.

Subsequent updates will probably be available in softcopy before they are available in hardcopy. This means that at any time from the availability of a release, softcopy versions should be regarded as the most up-to-date.

For CICS Transaction Server 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.

Updates to the softcopy are clearly marked by revision codes (usually a “#” character) to the left of the changes.

---

## Bibliography

### CICS Transaction Server for OS/390

<i>CICS Transaction Server for OS/390: Planning for Installation</i>	GC33-1789
<i>CICS Transaction Server for OS/390 Release Guide</i>	GC34-5352
<i>CICS Transaction Server for OS/390 Migration Guide</i>	GC34-5353
<i>CICS Transaction Server for OS/390 Installation Guide</i>	GC33-1681
<i>CICS Transaction Server for OS/390 Program Directory</i>	GI10-2506
<i>CICS Transaction Server for OS/390 Licensed Program Specification</i>	GC33-1707

### CICS books for CICS Transaction Server for OS/390

#### General

<i>CICS Master Index</i>	SC33-1704
<i>CICS User's Handbook</i>	SX33-6104
<i>CICS Transaction Server for OS/390 Glossary (softcopy only)</i>	GC33-1705

#### Administration

<i>CICS System Definition Guide</i>	SC33-1682
<i>CICS Customization Guide</i>	SC33-1683
<i>CICS Resource Definition Guide</i>	SC33-1684
<i>CICS Operations and Utilities Guide</i>	SC33-1685
<i>CICS Supplied Transactions</i>	SC33-1686

#### Programming

<i>CICS Application Programming Guide</i>	SC33-1687
<i>CICS Application Programming Reference</i>	SC33-1688
<i>CICS System Programming Reference</i>	SC33-1689
<i>CICS Front End Programming Interface User's Guide</i>	SC33-1692
<i>CICS C++ OO Class Libraries</i>	SC34-5455
<i>CICS Distributed Transaction Programming Guide</i>	SC33-1691

<i>CICS Business Transaction Services</i>	SC34-5268
---	-----------

#### Diagnosis

<i>CICS Problem Determination Guide</i>	GC33-1693
<i>CICS Messages and Codes</i>	GC33-1694
<i>CICS Diagnosis Reference</i>	LY33-6088
<i>CICS Data Areas</i>	LY33-6089
<i>CICS Trace Entries</i>	SC34-5446
<i>CICS Supplementary Data Areas</i>	LY33-6090

#### Communication

<i>CICS Intercommunication Guide</i>	SC33-1695
<i>CICS Family: Interproduct Communication</i>	SC33-0824
<i>CICS Family: Communicating from CICS on System/390</i>	SC33-1697
<i>CICS External Interfaces Guide</i>	SC33-1944
<i>CICS Internet Guide</i>	SC34-5445

#### Special topics

<i>CICS Recovery and Restart Guide</i>	SC33-1698
<i>CICS Performance Guide</i>	SC33-1699
<i>CICS IMS Database Control Guide</i>	SC33-1700
<i>CICS RACF Security Guide</i>	SC33-1701
<i>CICS Shared Data Tables Guide</i>	SC33-1702
<i>CICS Transaction Affinities Utility Guide</i>	SC33-1777
<i>CICS DB2 Guide</i>	SC33-1939

### CICSplex SM books for CICS Transaction Server for OS/390

#### General

<i>CICSplex SM Master Index</i>	SC33-1812
<i>CICSplex SM Concepts and Planning</i>	GC33-0786

<i>CICSplex SM User Interface Guide</i>	SC33-0788
<i>CICSplex SM Web User Interface Guide</i>	SC34-5403
<i>CICSplex SM View Commands Reference Summary</i>	SX33-6099
<b>Administration and Management</b>	
<i>CICSplex SM Administration</i>	SC34-5401
<i>CICSplex SM Operations Views Reference</i>	SC33-0789
<i>CICSplex SM Monitor Views Reference</i>	SC34-5402
<i>CICSplex SM Managing Workloads</i>	SC33-1807
<i>CICSplex SM Managing Resource Usage</i>	SC33-1808
<i>CICSplex SM Managing Business Applications</i>	SC33-1809
<b>Programming</b>	
<i>CICSplex SM Application Programming Guide</i>	SC34-5457
<i>CICSplex SM Application Programming Reference</i>	SC34-5458
<b>Diagnosis</b>	
<i>CICSplex SM Resource Tables Reference</i>	SC33-1220
<i>CICSplex SM Messages and Codes</i>	GC33-0790
<i>CICSplex SM Problem Determination</i>	GC33-0791

## Other CICS books

<i>CICS Application Programming Primer (VS COBOL II)</i>	SC33-0674
<i>CICS Application Migration Aid Guide</i>	SC33-0768
<i>CICS Family: API Structure</i>	SC33-1007
<i>CICS Family: Client/Server Programming</i>	SC33-1435
<i>CICS Family: General Information</i>	GC33-0155
<i>CICS 4.1 Sample Applications Guide</i>	SC33-1173
<i>CICS/ESA 3.3 XRF Guide</i>	SC33-0661

If you have any questions about the CICS Transaction Server for OS/390 library, see *CICS Transaction Server for OS/390: Planning for Installation* which discusses both hardcopy and softcopy books and the ways that the books can be ordered.

---

## Summary of changes

Changes are marked by vertical lines to the left of the changes.

---

### Changes for the CICS Transaction Server for OS/390 Release 3 edition

This edition is based on the *Operations and Utilities Guide* for CICS Transaction Server for OS/390 Release 2, SC33-1685-01.

For CICS Transaction Server for OS/390 Release 3, the following changes have been made:

- New section on the Local catalog storage program, DFHSMUTL.
- PROCESS command.
- Add REMOVE option to DELETE.
- Changes to DFHSTUPE and DFH\$MOLS.

# Additional information for this release includes:

- # • “Chapter 9. Recovery statistics program (DFH\$STER)” on page 91
- # • “Chapter 15. Load module scanner (DFHEISUP)” on page 177

# and

---

### Changes for the CICS Transaction Server for OS/390 Release 2 edition

This edition is based on the *Operations and Utilities Guide* for CICS Transaction Server for OS/390 Release 1, SC33-1685-00.

For CICS Transaction Server for OS/390 Release 2, the following changes have been made:

- Information about sizing DASD-only log streams has been added to “Chapter 6. Log stream sizing migration utility, DFHLSCU” on page 41.
- Information about the CICS diagnostic run mechanism has been added to “Chapter 22. Recovery manager utility program (DFHRMUTL)” on page 223.
- Information about the new BMS macro generation utility program (DFHBMSUP) has been added in “Chapter 23. BMS macro generation utility program (DFHBMSUP)” on page 229.
- The statistics utility program, DFHSTUP, produces the following additional statistics reports:
  - DB2 connections
  - DB2 entries.

See “Chapter 8. Statistics utility program (DFHSTUP)” on page 77.

- Three new resource types have been added to the DFHCSDUP utility program as follows:
  - DB2CONN
  - DB2ENTRY
  - DB2TRAN.

See “Chapter 13. System definition file utility program (DFHCSDUP)” on page 139.

- The SCAN command has been added to the DFHCSDUP utility program. See “Chapter 13. System definition file utility program (DFHCSDUP)” on page 139.

---

## Changes for the CICS Transaction Server for OS/390 release 1

The main changes made to this book for CICS Transaction Server for OS/390 release 1 are summarized below:

### **DFHCESD sample shutdown assist program**

A new sample, designed to be invoked at normal or immediate shutdown; it enables CICS to shut down in a controlled manner, within a reasonable period of time.

### **DFHRMUTL Recovery Manager utility program**

A new utility program to override automatic startup settings and improve performance at initial and cold starts.

### **Transaction Affinities Utility**

A utility program designed to detect transaction affinity in application programs.

### **DFH\$OFAR Offsite Automatic Reply program**

A NETVIEW exec that assists in the disaster recovery of a CICSplex when data sets have been used in RLS mode and OFFSITE=YES has been specified as a system initialization parameter.

### **DFHLSCU Log Stream Sizing utility**

A new utility program to assist in the sizing of log stream structures in the MVS coupling facility.

The chapter on DFHJUP has been amended to describe how to access, format, and print journaled data in CICS log streams, and on SMF data sets.

See the *CICS/ESA 4.1 Operations and Utilities Guide* for information on XRF.

In addition, Appendix A, about operating procedures, has been removed because it is no longer current.

---

## 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. Operating CICS from a console device” on page 29
- “Chapter 5. How to shut down CICS” on page 35.

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

- CICS startup and CICS initialization parameters in the *CICS System Definition Guide* for information about CICS system definitions; including CICS startup JCL and system initialization parameters
- The *CICS Recovery and Restart Guide*, for information about CICS recovery and restart
- The *CICS Supplied Transactions* manual, for information about the master terminal transactions provided by CICS
- The *CICS IMS Database Control Guide*, for information about operating CICS with IMS/ESA® database control (DBCTL)
- The *CICS DB2 Guide*, for information about operating CICS with DB2®
- 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 read the *CICSplex SM Concepts and Planning* .

### 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 should record actions taken while operating CICS.



---

## Chapter 1. Overview of CICS operations

This chapter provides an overview of CICS operations. It describes:

- “Starting up CICS”
- “Controlling CICS operation” on page 10
- “Shutting down CICS” on page 12.

Details about CICS operations are given in “Chapter 2. Starting up CICS regions” on page 17 through Chapter 5.

---

### 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 Recovery and Restart Guide*.

### The types of CICS startup

CICS startup can be any of the following types:

## Startup type

### Effect

- Initial** CICS starts with no reference to any system activity recorded in the CICS global catalog and system log from a previous run of CICS. For more information, see “CICS actions on an initial start”.
- Cold** CICS starts with limited reference to any system activity recorded in the CICS global catalog and system log from a previous run of CICS. For more information, see “CICS actions on a cold start” on page 5.
- Warm** CICS starts, 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 starts, 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, the type of startup (and therefore the actions it takes) depends primarily on the following:

- The value of the START system initialization parameter
- Two records in the CICS global catalog:
  - The recovery manager control record
  - The recovery manager autostart override record.

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

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 System Definition Guide*.

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

## CICS actions on an initial start

The CICS global catalog and system log are initialized, and all information in them is lost. **Because resynchronization information for remote systems is not preserved, damage may be done to distributed units of work.**

It should rarely be necessary to perform an initial start. Examples of times when an initial start is necessary are:

- When bringing up a new CICS system for the first time
- After a serious software failure, when the global catalog or system log has been corrupted.

## 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. With the exception of resynchronization information for remote systems noted below, 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:

- 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.
- Transient data queue (TDQUEUE) definitions are purged from the catalog.
- File control records are purged from the catalog.
- Resource definition information is obtained as follows:
  - Tables specified by system initialization parameters, such as MCT=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.
- Resynchronization information relating to remote systems or to RMI-connected resource managers is preserved. The CICS system log is scanned during startup, and information regarding unit of work obligations to remote systems, or to non-CICS resource managers (such as DB2) connected through the RMI, is preserved. (That is, any decisions about the outcome of local UOWs, needed to allow remote systems or RMI resource managers to resynchronize their resources, are preserved.)

However, note that recovery information for remote systems connected by LU6.1 links, or for earlier releases of CICS systems connected by MRO is *not* preserved.

- The journal DFHLOG and DFHSHUNT entries in the catalog are used, and all other journals and journal models are purged.

## 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:

- Resource definition information is obtained as follows:
  - Tables specified by system initialization parameters, such as MCT=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.

**Note:** The DCT load module is not referenced during a warm start, and any setting on the DCT system initialization parameter is ignored.

- 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
  - High-water mark number of the unit of recovery descriptor.
- The following pieces of information relating to logically recoverable, physically recoverable and non-recoverable intrapartition transient data queues are restored:
  - All data defining the queues. This information is restored from the global catalog, including trigger level information, ATI transaction IDs, ATI terminal IDs and so on.
  - All state-related data. This information is retrieved from the warm keypoint which was written to the log, including:
    - Record count
    - Read pointer value
    - Write pointer value
    - Information about whether or not a trigger transaction has been attached.

All intrapartition transient data queues are installed as ENABLED. Trigger transactions are rescheduled if required.

Extrapartition transient data queues are opened if OPEN=INITIAL is specified in the queue definition.

- 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 DSNNAME.
- 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.
- 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 (CMSG).
  - 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.
- Transaction and system dump table options are held in the global catalog and reapplied at a warm start.
- Journals and journal models are restored from the catalog.

## 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, Recovery Manager collects all of the log records pertaining to in-flight tasks. It acquires locks on any records that they updated and restores the tasks as shunted UOWs, to be backed out after initialization 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<sup>®</sup> 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 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 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 Concepts and Planning*, GC33-0786.

## 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 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 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 39.

---

## 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 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. If the CESD default shutdown transaction is enabled, existing tasks are given a short time to finish before they are purged.  
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 after 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 phases. (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.
- Tasks that already exist will complete. (Long running tasks, such as conversational tasks, must end before this stage of shutdown 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.
- Unless SDTRAN=NO or NOSDTRAN was specified, the shutdown task starts the specified shutdown transaction (default is CESD). CESD manages the purging of long-running user tasks.
- 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. All currently open CICS files are now closed.
3. The transient data CI buffer and the temporary storage buffers are flushed.
4. 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.

- A warm-start-possible indicator. This status applies on the next initialization of CICS if START=AUTO is specified.
5. Transient data is terminated.
  6. A dump is taken, if one is required.
  7. 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).
  8. The local and global catalogs are closed.
  9. The following message is issued:  
DFHKE1799 *applid* TERMINATION OF CICS/ESA® IS COMPLETE
  10. 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 for any kind of shutdown. They are just given less time for immediate shutdown before being purged.
- None of the programs listed in the shutdown PLT is 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 quiesce 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. Unless SDTRAN=NO or NOSDTRAN was specified, the shutdown task starts the specified shutdown transaction (the default is CESD). CESD manages the purging of long-running user tasks.

### **Second stage of immediate shutdown**

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

1. Transient data is terminated.
2. A dump is taken, if requested.
3. Interregion sessions are terminated.
4. If CICS is signed on to the CICS availability manager (CAVM), a “signoff abnormal” request is made from CAVM.
5. The local catalog and global catalog are left to be closed by the operating system.
6. The following message is issued:  
DFHKE1799 *applid* TERMINATION OF CICS/ESA IS COMPLETE
7. 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 System Definition Guide*. For example, that book describes the system initialization parameters in detail, and describes how to create a CICS startup job stream.

---

### Starting CICS

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, after 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 20.)

---

### 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=DFHSIP 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 whether to perform an initial, cold, warm, or emergency start by inspecting two records in the global catalog:

- The recovery manager control record

- The recovery manager autostart override record.

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

#### START=INITIAL

The new run of CICS has no reference to any previous run. The global catalog and system log are initialized, and all information in them is lost.

#### START=COLD

The new run of CICS has limited reference to the previous run, and uses the same global catalog and system log. In particular, resynchronization information needed by remote systems to resynchronize their units of work is preserved.

#### START=STANDBY

CICS starts up as an XRF alternate CICS region, by initializing only to the point at which it can monitor the active CICS region. Depending on how the active CICS region was shut down, the alternate CICS region completes either a warm or emergency restart, if it needs to take over, as follows:

- If the active CICS region was shut down via a successfully completed CEMT PERFORM SHUTDOWN TAKEOVER command, the alternate CICS region performs a **warm** start.
- If the active CICS region was shut down abnormally, the alternate CICS region performs an **emergency** restart.

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

For example, if your CICS startup procedure specifies:

```
//INITCICS EXEC PGM=DFHSIP,REGION=&REG,
//          PARM=('CLONE=HT##,SYSIDNT=HTH1,SIT=6$,SYSIN,CN')
//*
//SYSIN DD DISP=SHR,DSN=&libpfx..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 member CICSH### of the CICSTS13.CICS.CICSH###.SYSIN data set
3. The system console.

---

## 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 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
//*****
//*
//CICS530 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, CICSTS13.CICS.SYSIN, that has system initialization parameters unique to each *type* of CICS region.

For information about the CICS-supplied startup procedure, see the *CICS Transaction Server for OS/390 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 *OS/390 MVS System Commands*.

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 CICS530
```

```
//CICS530 PROC /* //DUMMY EXEC PGM=IEFBR14 /* // START
CICSTASK.CICSHTH1,SYSIDNT='HTH1',CLONE='HT##' /*      START=COLD // START
CICSTASK.CICSHAH1,SYSIDNT='HAH1',CLONE='HA##' /*      START=COLD // START
CICSTASK.CICSHAH2,SYSIDNT='HAH2',CLONE='HA##' /*      START=COLD // START
CICSTASK.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, CICSTS13.CICS.SYSIN, that has system initialization parameters unique to each *type* of CICS region.

For information about the CICS-supplied startup procedure, see the *CICS Transaction Server for OS/390 Installation Guide*.

If you are running CICS with RACF<sup>®</sup>, 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 RACF Security Guide*.

---

## Overriding system initialization parameters during startup

After 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 Transaction Server for OS/390 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 on page 22, because the SYSIN data is displayed on the console.

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

The notes after Figure 3 identify specific messages of interest.

```

JES2 JOB LOG -- SYSTEM M V S H -- NODE W I N M V S H

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, 1997
14.55.13 . $HASP373 CIDCITH1 STARTED - INIT 2 - CLASS C - SYS M V S H
14.55.13 . IEF403I CIDCITH1 - STARTED - TIME=14.55.13
14.55.14 . 1 DFHPA1102 DBDCCICS READING OVERRIDE PARAMETERS FROM SYSIN.
14.55.14 . DFHPA1927 DBDCCICS * *****
14.55.14 . DFHPA1927 DBDCCICS * CICS system initialization parameters common to all clone TORs*
14.55.14 . DFHPA1927 DBDCCICS * *****
14.55.14 . 2 DFHPA1927 DBDCCICS APPLID=CICSHTH1 The CICS application identifier
14.55.14 . 3 DFHPA1927 DBDCCICS CICSSVC=218 The default CICS SVC number
14.55.14 . 4 DFHPA1927 DBDCCICS CWAKEY=CICS CICS key for the CWA
14.55.14 . DFHPA1927 DBDCCICS DBP=1$ Dynamic backout program - no local DL/I support
14.55.14 . DFHPA1927 DBDCCICS DCT=NO RDO-generated TD resources are preferred
14.55.14 . 5 DFHPA1927 DBDCCICS DFLTUSER=IVPUSER Default userid is IVPUSER
14.55.14 . 6 DFHPA1927 DBDCCICS DTRPGM=WLM DYP 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 5.3 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 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
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 5.3.0 Startup is in progress.
14.55.15 . +DFHXS1100I CICSHTH1 Security initialization has started.
14.55.15 . +DFHSI1501I CICSHTH1 Loading CICS nucleus.
14.55.16 . +DFH DU0304I CICSHTH1 Transaction Dump Data set DFHDMPA opened.
14.55.16 . +DFHXS1102I CICSHTH1 Security is inactive.
14.55.16 . +DFHXS1101I CICSHTH1 Security initialization has ended.

```

Figure 3. Console messages for startup of TOR - CICSHTH1 1/2

```

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 . +DFHAI0101I CICSHTH1 AITM initialization has started.
14.55.20 . +DFHFC0101I CICSHTH1 File Control initialization has ended.
14.55.20 . +DFHTD0101I CICSHTH1 Transient Data initialization.
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 4. Console messages for startup of TOR - CICSHTH1 2/2

#### 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 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 an initial or cold start; at 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 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 Storage requirements for a CICS region, DSALIM, and DFHSIT, the default system initialization table in the *CICS 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) as a system initialization parameter, 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 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 CPCs<sup>1</sup>
- 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 Transaction Server for OS/390 Release 3 region running in an MVS image can communicate with a CICS/VSE<sup>®</sup> partition running under VSE/ESA. Alternatively, on two MVS platforms, the communicating systems could be CICS at one end and IMS<sup>™</sup> at the other.

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.

---

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.

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 ix.

Intersystem communication and Multiregion operation in the *CICS Intercommunication Guide* contain information about CICS intercommunication, and how to implement both intersystem communication and multiregion operation. This chapter assumes that you are familiar with the CICS intercommunication facilities.

---

## Enabling MRO

To be able to use CICS MRO, you must first install support for MRO, as described in the *CICS 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 External Interfaces Guide* manual.



---

## Chapter 4. Operating CICS from a console device

You can operate CICS from a **console device**<sup>2</sup>. 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 region in an XRF environment, you must use the CEBT command, and issue it from a console device.

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 names.

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 30.

For information about defining console devices to CICS, see the *CICS System Definition Guide*. For further information about defining console devices to MVS, see the *OS/390 MVS Initialization and Tuning Guide*.

---

### 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

---

2. A console device can be a locally-attached system console, a TSO user defined as a console, or an automated process such as Netview.

- 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 32.

## Entering commands from TSO

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 `CONSNAME(INTERNAL)`, 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.

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 Transaction Server for OS/390 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:

```
DFHSI1552 applid Restart error reported above. Reply 'GO' or 'CANCEL'.
```

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 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, which 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 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 21.

## Sample console messages for CICS shutdown

Sample console messages issued when CICS starts up are given in “Chapter 5. How to shut down CICS” on page 35.



---

## Chapter 5. How to shut down CICS

This chapter describes how to shut down CICS. For an overview of CICS shutdown, see “Shutting down CICS” on page 12.

To shut down 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 produces a dynamic storage dump after shutdown has completed.

#### PLT(xx)

CICS runs 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 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.

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:

```
13.04.37 JOB08579 +DFHTM1715 IYK4ZEE1 CICS/ESA is being quiesced by userid CICSUSER
                    in transaction CEMT at netname IG2S66B9.
13.04.37 JOB08579 +DFHDM0102I IYK4ZEE1 CICS is quiescing.
13.04.37 JOB08579 +DFHCESD IYK4ZEE1 SHUTDOWN ASSIST TRANSACTION CESD STARTING.
                    SHUTDOWN IS NORMAL.
13.04.37 JOB08579 +DFHTM1781 IYK4ZEE1 CICS shutdown cannot complete because some
                    non-system user tasks have not terminated.
13.06.37 JOB08579 +DFHCESD IYK4ZEE1 THERE ARE NOW 0002 TASKS STILL IN THE SYSTEM.
13.06.54 JOB08579 +DFHDM0303I IYK4ZEE1 Transaction Dump Data set DFHDMPA closed.
13.06.54 JOB08579 +DFHCESD IYK4ZEE1 PURGING TRANID CECI, TERMID 66B8, USERID CICSUSER,
                    TASKNO 000026
13.06.54 JOB08579 +DFHKE0030 - Abend ---/ATCH in Program DFHPCP Entry Point 06E08C80.
13.06.56 JOB08579 +DFHCESD IYK4ZEE1 THERE ARE NOW 0001 TASKS STILL IN THE SYSTEM.
13.06.59 JOB08579 +DFHTM1782I IYK4ZEE1 All non-system tasks have been successfully
                    terminated.
13.06.59 JOB08579 +DFHZC2305I IYK4ZEE1 Termination of VTAM sessions beginning
13.07.01 JOB08579 +DFHZC2316 IYK4ZEE1 VTAM ACB is closed
13.07.03 JOB08579 +DFHRM0204 IYK4ZEE1 There are no indoubt, commit-failed or
                    backout-failed UOWs.
13.07.04 JOB08579 +DFHRM0130 IYK4ZEE1 Recovery manager has successfully quiesced.
13.07.07 JOB08579 +DFHKE1799 IYK4ZEE1 TERMINATION OF CICS/ESA IS COMPLETE.
```

---

## Shutting down CICS immediately

To shut down CICS immediately, use the CEMT PERFORM SHUTDOWN IMMEDIATE command.

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 CICSHTH1 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
16.16.01 . +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 the *CICS/ESA 4.1 Operations and Utilities Guide*.



---

## 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:

Table 1. CICS utility programs

Name	Utility program use	See page
DFHLSCU	Log stream and coupling facility sizing utility	41
DFHJUP	Journal select, print, and copy	53
DFHSTUP	Statistics formatting	77
DFH\$STER	Recovery statistics program	91
DFHTU530	Trace formatting	93
DFHDU530	Transaction dump formatting	103
DFHMNDUP	Monitoring dictionary record creation	119
DFH\$MOLS	Sample monitoring formatting	124
DFHCSDUP	System definition data set utility	139
DFHMSCAN	Macro-level programs identification	191
DFHEISUP	Load module scanner	177
DFHSNMIG	Signon table to RACF migration	195
DFH\$STED	Sample stagger end-of-day time	197
DFHMEU	Message editing	199
DFHCESD	Sample shutdown assist program	217
DFHRMUTL	Recovery Manager batch utility	223
DFHBMSUP	BMS macro generation utility	229
DFH\$OFAR	Offsite Automatic Reply program	235
DFHSMUTL	Local catalog storage manager domain subpool record manipulation	237

**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 described in the *CICS Transaction Affinities Utility Guide*.



---

## Chapter 6. Log stream sizing migration utility, DFHLSCU

If you are migrating to CICS Transaction Server for OS/390 Release 3 from CICS/ESA 4.1 or CICS/ESA 3.3, you will need, for the first time, to define one or both of the following types of CICS log stream:

- Log streams that use coupling facility (CF) structures
- CICS Transaction Server for OS/390 Release 3 DASD-only log streams.

For advice on which log streams benefit from using coupling facility structures, and which DASD-only logging, see the *CICS Transaction Server for OS/390 Installation Guide*.

---

### Output from DFHLSCU

DFHLSCU is a migration utility that helps you to define your CICS Transaction Server for OS/390 Release 3 log streams, based on your CICS/ESA 4.1 or CICS/ESA 3.3 journaling activity. It examines one or more CICS/ESA 4.1 or CICS/ESA 3.3 journal data sets, and produces a report. The conclusion of the report is divided into two sections:

#### 1. CF log stream:

The first section of the report's conclusion assumes that you want to define a log stream that uses a coupling facility structure. It contains recommended values for:

##### **AVGBUFSIZE**

The average buffer size, in bytes, of a log stream structure in the coupling facility. It is important, particularly in MVS releases before OS/390 Release 3, that the value you specify for AVGBUFSIZE reflects as accurately as possible the real size of most log blocks written to the structure. This leads to efficient use of the space in the coupling facility and minimum DASD offloading frequency. This is less important in OS/390 Release 3 and later, because OS/390 performs some dynamic tuning.

##### **INITSIZE**

The initial amount of space, in kilobytes, to be allocated for the log stream structure in the coupling facility. You define this attribute in your CFRM policy.

##### **HIGHOFFLOAD**

The point in primary storage (that is, in either the coupling facility structure or the staging data set), as a percentage of space consumed, at which the MVS system logger starts its offload process—which can cause data from the log stream to be offloaded to log stream DASD data sets. (The offload process is described in the *CICS Transaction Server for OS/390 Installation Guide*.) You define this attribute in your DEFINE LOGSTREAM job.

##### **LOWOFFLOAD**

The point in primary storage, as a percentage of space consumed, at which the MVS system logger stops offloading data from the log stream to log stream DASD data sets. You define this attribute in your DEFINE LOGSTREAM job.

**SIZE** The maximum size, in kilobytes, of the log stream structure in the coupling facility. You define this attribute in your CFRM policy.

**STG\_SIZE**

Optionally, a staging data set can be used with a CF log stream. For guidance about when to use a staging data set, see the *CICS Transaction Server for OS/390 Installation Guide*. STG\_SIZE is the size, as a number of 4K blocks, of the staging data set, if one is required. You define this attribute in your DEFINE LOGSTREAM job.

**2. DASD-only log stream:**

The second section of the report's conclusion assumes that you want to define a log stream that uses DASD-only logging. It contains recommended values for the following attributes (all of which are defined in your DEFINE LOGSTREAM job):

**HIGHOFFLOAD**

The point in primary storage (that is, in the staging data set), as a percentage of space consumed, at which the MVS system logger starts its offload process.

**LOWOFFLOAD**

The point in primary storage as a percentage of space consumed, at which the MVS system logger stops offloading data from the log stream to log stream DASD data sets.

**MAXBUFSIZE**

The size, in bytes, of the largest block of data that can be written to the log stream.

**STG\_SIZE**

The size, as a number of 4K blocks, of the staging data set for the log stream.

A DASD-only log stream always uses a staging data set.

The values provided by DFHLSCU are **estimates** and may not match your actual experience. In particular, they may be affected by changes in the pattern of logging such as:

- RLS file usage moving logging from an FOR into AORs.
- Combining journals onto a single log stream.
- Introducing more cloned AORs.

While the recommended values provide a starting point for structure sizing, you should monitor actual usage and adjust it as required.

---

## Job control statements to run the DFHLSCU program

DFHLSCU runs as a standard operating system job. You must define a JOB statement, and EXEC statement, and a JOURNAL statement. The job stream to run the DFHLSCU utility should include the following DD statements:

### STEPLIB DD

defines a partitioned data set (DSORG=PO) containing the DFHLSCU module. If the module is in a library in the link list, this statement is not required.

### JOURNAL DD

defines the CICS/ESA 4.1, or CICS/ESA Version 3, journal data set that is to be examined by the utility. Multiple journal data sets can be specified by concatenating additional data sets to the DD statement, they should originate from the same CICS system (that is, they should not be a mixture from different regions).

See the *CICS/ESA 4.1 Operations Guide* for information about journal data set definitions in JCL and “Considerations when using DFHLSCU” on page 45.

### SYSPRINT DD

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

defines values and parameters to be used by the utility. This file must be in 80-byte record format. One SYSIN statement per line is permitted. Ensure that your statements do not exceed column 71.

Using SYSIN statements, you can pass values to the utility to be used in the report calculations and recommendations. These assume default values if you do not specify them explicitly.

The utility is invoked as follows:

```
//*****  
//* RUN DFHLSCU (LOGSTREAM CALCULATIONS UTILITY).  
//*  
//*  
//*****  
//LSCU EXEC PGM=DFHLSCU  
//STEPLIB DD DSN=CICSTS13.CICS.SDFHLOAD,DISP=SHR  
//*****  
//* CICS journal name(s)  
//*****  
//JOURNAL DD DISP=SHR,DCB=RECFM=VB,  
// DSN=CICSLLOG  
//*****  
//* Output data will go to SYSPRINT  
//*****  
//SYSPRINT DD SYSOUT=A,DCB=RECFM=FBA  
//SYSIN DD *  
JNLTYPE( )  
INTERVAL( )  
AKPFREQ( )  
LOGSNUM( )  
TRANSDUR( )  
/*  
/*
```

Figure 5. Skeleton JCL to run DFHLSCU

---

## SYSIN control statements for the DFHLSCU utility

You can use SYSIN statements to provide further input to DFHLSCU, and to tailor the report that it provides.

### Format of the SYSIN control statements

```
SYSIN DD *  
[JNLTYPE(SYSTEM|FWDREC|USRJNL)]  
[INTERVAL(minutes)]  
[AKPFREQ(data-value)]  
[LOGSNUM(data-value)]  
[TRANSDUR(seconds)]
```

Figure 6. SYSIN control statements for the DFHLSCU program

If you do not define a SYSIN data set, or SYSIN does not contain any control statements, default values are assumed. Each control statement must be on a separate line and must contain no spaces. The SYSIN statements you can code are as follows.

#### JNLTYPE

This statement indicates the type of journal the data set represents. Code this statement with one of the following operands:

##### SYSTEM

This data set represents the system log.

##### FWDREC

This data set represents a forward recovery log.

## **USRJNL**

This data set represents a user journal or autojournal.

The default is SYSTEM.

## **INTERVAL**

This statement permits you to divide the data set into time blocks, each time block being summarized in the summary report. This provides you with information allowing you to analyze journaling behavior over different time segments.

Specify this statement in minutes (0 thru 999999). The default is 30. A value of 0 indicates that no time-segmenting is to occur and that the period covered by the entire data set is to be used.

## **AKPFREQ**

This statement specifies the activity keypoint frequency. This is relevant only for calculations of space needed by system logs. It is used to calculate the size of the CF space or staging data set needed for the system log. The value here is the AKPFREQ that you intend to use at CICS Transaction Server for OS/390 Release 3.

Code either a value of 0, or a value in the range 200 thru 65535. The default is 4000.

## **LOGSNUM**

For a CF log stream, this statement specifies the number of log streams that can use the structure associated with this journal or log. It is used in the calculation of the INITSIZE and SIZE attributes to include in the CFRM policy.

Code a value in the range 1 through 512. The recommended range is 1 through 20. The default is 10.

## **TRANSDUR**

This statement specifies the transaction duration which is the execution time (between syncpoints) of the longest-running transaction that runs as part of the normal workload. TRANSDUR is only relevant for system log calculations.

Specify this value in seconds (0 thru 999999). The default is 3.

---

## **Considerations when using DFHLSCU**

The results from DFHLSCU will vary depending upon the values passed on SYSIN, and on the journal record activity found on the target log.

DFHLSCU divides up the target log into time segments according to the INTERVAL parameter. It analyses the log records found in each segment and determines which is the busiest segment, based on this logging activity. This segment is then used to determine the parameters for your logstream definitions and space requirements.

A system log from a CICS system with a consistent workload will have a reasonably regular time period between activity keypoints. Conversely, a system log from a CICS system with irregular workloads, that rise and fall in no consistent way, or which rise to a peak then drop back to a low level, or else which rise to a plateau for a period of time before dropping back once more, will all result in varying time periods between activity keypoints.

Since DFHLSCU bases its calculations upon the time period which contains the busiest workload (that is, which generated most log records), the INTERVAL

parameter will have a potentially marked effect upon the results generated by the utility if the CICS run which produced the target log had an inconsistent workload. For example, consider a target log from a ten hour CICS run. During that run, the system was lightly used for all but a one hour plateau near the middle of the run, when the workload rose rapidly to a much higher value, and many CICS log records were generated during that hour.

If DFHLSCU were run against such a log, with an INTERVAL parameter of considerably greater time than the duration of this plateau, then the results produced by the utility would be misleading. This is since DFHLSCU has to average the effect from each time segment, and the segments will span part of the plateau and also parts of the surrounding periods of low workload.

A general guide to specifying a value for the INTERVAL parameter is as follows. Firstly, study the workload of the CICS run which generated the target log for the utility. If the workload is reasonably consistent with no large peaks, troughs or plateaus, differing values for INTERVAL should have a limited effect. However, if the workload varies considerably, determine the duration of the busiest period of the CICS workload and specify an INTERVAL parameter of half this value. In this way, DFHLSCU should divide the log into time segments such that one segment should reside completely within a time period of consistently busy log activity. For the example with a one hour busy period given above, this methodology yields an INTERVAL value not greater than thirty minutes.

The recommendation is that DFHLSCU be run against a journal or set of journals that were produced from a single CICS region. DFHLSCU then outputs recommended values for structure sizing based on that CICS region. If the structure is intended to accommodate logstreams from more than one CICS region, the recommended approach is to run DFHLSCU against journals from each individual CICS region, and then sum the sizing recommendations from each output to determine the final structure sizing requirements.

**Note:** An additional 300,000 bytes should be added to the size value used for the structure, DFHLSCU does not reflect this in the recommended output values because it is added by MVS when defining the structure.

| If DFHLSCU is run against a target journal that was defined with FORMAT=SMF,  
| an IEC036I 002-04 abend occurs when the first record is read. This is because  
| DFHLSCU does not support analysis of target journals in SMF format.

| If you want the target journal to be directed to a log stream when migrating to  
| CICS Transaction Server for OS/390, you must redefine it under CICS/ESA so that  
| it does not use SMF formatting. Run a typical workload against the redefined  
| journal before using DFHLSCU to produce a report for it.

| If you want the target journal to be directed to SMF when migrating to CICS  
| Transaction Server for OS/390, DFHLSCU does not give any benefit so do not use  
| it against the SMF-format journal.

| Unlike CICS/ESA, CICS Transaction Server for OS/390 does not support the  
| journaling of SMF-format log data to non-SMF media (that is, the CICS Log  
| Manager does not write records in SMF format to a log stream).

---

## DFHLSCU return codes

The following errors can occur in DFHLSCU:

**Return code****Error description**

- 04 There are duplicate SYSIN statements.
- 08 This return code can be issued for either of the following reasons:
- A syntax error is detected in a SYSIN record.
  - An illegal value has been detected in a SYSIN record.
- 12 This return code can be issued for any of the following reasons:
- There is no JOURNAL DD statement.
  - The journal data set has failed to open.
  - There has been a runtime error in the journal data set.
  - SYSPRINT has failed to open.

---

## Output from the DFHLSCU utility

The DFHLSCU utility produces a report, directed to SYSPRINT. The report consists of a summary for every interval (as specified on the INTERVAL SYSIN statement). The end of the report provides conclusions and recommendations:

1. **For a CF log stream**, it provides:

- The largest calculated AVGBUFSIZE.
- A sample definition for the log stream structure. The definition contains recommended values for:
  - AVGBUFSIZE
  - MAXBUFSIZE
- A sample definition for CF space required for the log stream. The definition contains recommended values for:
  - INITSIZE
  - SIZE
- A sample log stream definition. The definition contains recommended values for:
  - HIGHOFFLOAD
  - LOWOFFLOAD
- For a CF log stream that will use duplexing, a recommended size for the staging data set.

You must edit the sample definitions and provide appropriate values for the structure name and prelist name.

2. **For a DASD-only log stream**, the report provides a sample log stream definition. The definition contains recommended values for the following attributes:

- HIGHOFFLOAD
- LOWOFFLOAD
- MAXBUFSIZE
- STG\_SIZE

**Note:** If you plan to define all your log streams as DASD-only, but there is a possibility that, at a later date, you might want to convert some of them to CF log streams, it is a good idea to save the output from DFHLSCU for later reference.

An example of a report produced by DFHLSCU, for estimating the size of a system log, is shown in Figure 7 on page 48.

\*\*\*\*\*  
 THE LOGSTREAM CALCULATIONS UTILITY REPORT.

JOURNAL TYPE : system log.  
 INTERVAL : 000002 minutes.  
 AKPFREQ : 00299  
 LOGSNUM : 0010  
 TRANSDUR : 000003 seconds.

\*\*\*\*\* REPORT SUMMARY INFORMATION \*\*\*\*\*

SEGMENT 00000001 **1**  
 DURATION 00006000 seconds **2**  
 TIME 17:12:52.9 **3**  
 DATE 1996.138 **4**  
 NUMBER OF BLOCKS : 00000002 **5**  
 WRITES PER SECOND : <1 **6**  
 AVERAGE RECORD SIZE : 00000065 **7**  
 AKP INTERVAL : 00179400 **8**

TYPE	QUANTITY	NUMBER OF BYTES	5.3 EQUIVALENT <b>9</b>
FC	00000000	0000000000000000	0000000000000000
JC	00000000	0000000000000000	0000000000000000
TD	00000000	0000000000000000	0000000000000000
TS	00000000	0000000000000000	0000000000000000
KP	00000000	0000000000000000	0000000000000000
RM	00000001	0000000000000066	0000000000000131
SP	00000009	0000000000000350	0000000000000524
Other	00000000	0000000000000000	*** NONE ***
Total	00000010		

From this, an AVGBUFSIZE of 00117 was calculated. **10**

\*\*\*\*\* REPORT SUMMARY INFORMATION \*\*\*\*\*

SEGMENT 00000002 DURATION 00000075 seconds  
 TIME 18:52:52.6 DATE 1996.138  
 NUMBER OF BLOCKS : 00000095  
 WRITES PER SECOND : 00000001  
 AVERAGE RECORD SIZE : 00000206  
 AKP INTERVAL : 00000195

TYPE	QUANTITY	NUMBER OF BYTES	5.3 EQUIVALENT
FC	00000090	0000000000011700	0000000000020520
JC	00000000	0000000000000000	0000000000000000
TD	00000003	0000000000000498	0000000000001461
TS	00000003	0000000000000150	0000000000000588
KP	00000000	0000000000000000	0000000000000000
RM	00000001	0000000000000066	0000000000000131
SP	00000018	0000000000000702	0000000000001050
Other	00000030	0000000000003900	*** NONE ***
Total	00000115		

From this, an AVGBUFSIZE of 00302 was calculated.

\*\*\*\*\*  
 The end of the JOURNAL data set has been reached.  
 \*\*\*\*\*

Figure 7. Sample output from DFHLSCU (Part 1 of 3)



```

#
#
#       If staging is to be used for this logstream, the following value
#       is that calculated for the staging data set size. This assumes the worst
#       case where only this logstream is actively connected to the structure. If
#       more log streams are to be connected in parallel, then this value should
#       be replaced by one obtained from dividing it by the number of streams.
#
#       STG_SIZE(4155) 16
#
#       *****
#       This section applies to DASD-only logstreams:- 17
#
#       You are recommended to complete the following definition and use it to
#       create a suitable logstream:
#
#       DATA TYPE(LOGR) REPORT(NO) 18
#       DEFINE LOGSTREAM NAME(userid.applid.DFHLOG)
#           DASDONLY(YES)
#           HIGHOFFLOAD(80)
#           LOWOFFLOAD(40)
#           STG_SIZE(1010)
#           MAXBUFSIZE(64000)
#
#

```

Figure 7. Sample output from DFHLSCU (Part 3 of 3)

#### Notes:

**1** This indicates the segment of the journal data set on which the utility is operating. Each segment holds journal records made over the period specified in the INTERVAL SYSIN statement, unless it is the final segment (when it contains journal records made between the end of the previous segment and the end of the journal data set).

**2** This indicates the time duration covered in the segment. It corresponds to the value on the INTERVAL parameter unless it is the final segment (when it is the time between the end of the previous segment and the end of the data set).

**3** This is the time at which the data being analyzed in the segment commenced its generation.

**4** This is the date on which the data being analyzed by DFHLSCU was generated. It is of the form *yyyy.ddd* where *yyyy* is the year, and *ddd* is the day within the year.

**5** The number of journal blocks contained in the segment is shown here.

**6** The average number of journal writes per second is shown here. Where the value is greater than 1, its value is shown as an integer value. Where the value is between 0 and 1, '< 1' is shown, and the estimates calculated by DFHLSCU (if based on this segment of the data set) will be inaccurate. Where it is 0, '0' is shown.

DFHLSCU's estimates are most accurate when the value for writes per second is high. (The maximum value is 25.)

**7** The average size of the records in the interval is shown here.

**8** The average time interval between activity keypoints is estimated from the information in the segment. This will only appear in the output if the JNLTYPE is SYSTEM.

**9** A section containing information about the specific records type found in the segment is provided. Record types included in this section are those generated by:

- File control (FC)
- Journal control (JC)
- Transient data (TD)
- Temporary storage (TS)
- Activity keypoint program (KP)
- Recovery manager (RM)
- Syncpoint program (SP)
- Other sources for which there is no equivalent on a CICS Transaction Server for OS/390 Release 3 log stream.

The 'QUANTITY' column shows the number of records of each type that was found. The total of the values in this column is given. The 'NUMBER OF BYTES' column shows the number of bytes that these records represent. The '5.3 EQUIVALENT' column shows the number of bytes, as calculated by DFHLSCU, that would be required at CICS Transaction Server for OS/390 Release 3 for an equivalent record.

**10** For each section, a value of AVGBUFSIZE is calculated.

**11** Start of the report's conclusion. The conclusion informs you which segment contained the most journaling activity and is based on the segment with the highest calculated value for AVGBUFSIZE.

**12** Start of the report's recommendations for log streams that use coupling facility structures.

**13** For a CF log stream, the recommended log stream structure definition to be included in your DEFINE STRUCTURE jobs.

**14** For a CF log stream, the recommended coupling facility space definition to be edited by you for inclusion in the CFRM policy.

**15** For a CF log stream, the recommended log stream definition. Note that the definition assumes that duplexing of data is not required. If duplexing is required, add to the definition:

```
STG_DUPLEX(YES) DUPLEXMODE(COND)
```

**16** For a CF log stream that uses duplexing, the recommended value for the STG\_SIZE attribute on your DEFINE LOGSTREAM job. This is the size of the staging data set required by the log stream. It does not take into account any other log streams that might be connected, at the same time, to the log stream structure. *If other log streams are to be connected, you should calculate a value for STG\_SIZE based on the value recommended by DFHLSCU, divided by the number of estimated connections.*

**17** Start of the report's recommendations for DASD-only log streams.

**18** For a DASD-only log stream, the recommended log stream definition.

**Additional Note**

The AKPFREQ and TRANSDUR information will only appear in the report if the JNLTYPE is SYSTEM. The total value in each summary section is that of the records which are relevant to the calculations that the utility carries out. It does not include records that can never occur in a 5.3 journal.

---

## Chapter 7. Using batch jobs to read log streams

You can run a batch job, such as DFHJUP, to read and process CICS log data in MVS system logger log streams and in MVS SMF data sets.

You can:

- Print or copy selected journal records from CICS log streams or SMF data sets, as specified by control statement input
- Select and print journal records on the basis of their sequential position in the log stream or SMF 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
- Print or copy an entire log stream or SMF data set.

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.

---

### **SUBSYS=(LOGR,DFHLGCNV,...) keyword**

If you are using a batch job to read log stream data, ensure that it includes the SUBSYS keyword as part of the DD statement for the logstream data set you are processing. The subsystem keyword for log streams specifies LOGR and exit routine parameters. For example,

```
SUBSYS=(LOGR,DFHLGCNV,...) keyword
```

**Note:** If you write your own batch program to process CICS log data sets using DFHLGCNV as the exit routine, your program must not open the data set more than once in a jobstep. If you do open the data set more than once in a jobstep, the following message is sent:

```
DFHLG0194 ERROR DETECTED BY CICS SUBSYSTEM.  ABEND  
IXG50 41 LOGGER SUBSYSTEM (LOGR) EXIT PROVIDED AN ERROR RETURN  
DD=TEST2  EXIT=DFHLGCNV  FUNCTION=OPEN  RETCODE=00000014
```

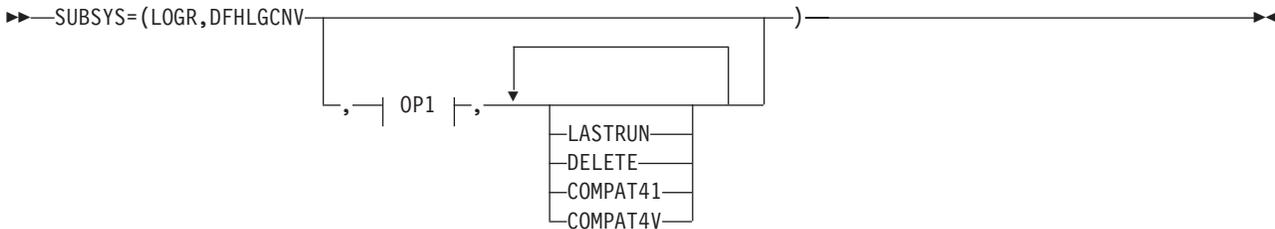
**SUBSYS=(LOGR,DFHLGCNV,...) keyword**

has the following form and syntax:

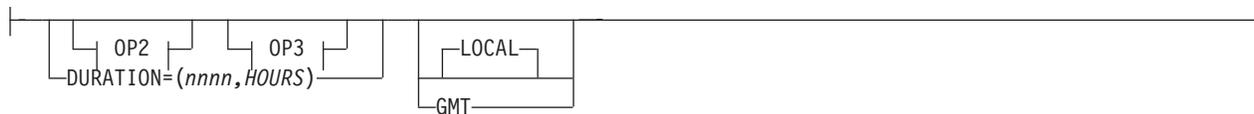
```
//ddname DD DSNAME=log_stream_name,
//      SUBSYS=
```

where SUBSYS expands as follows:

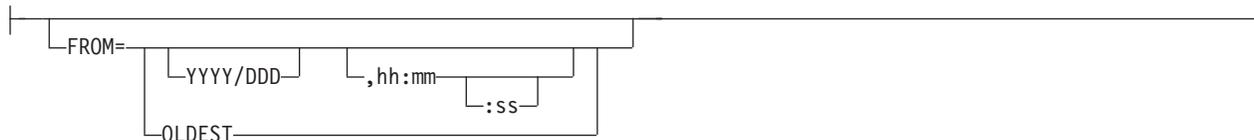
## SUBSYS



### OP1:



### OP2:



### OP3:

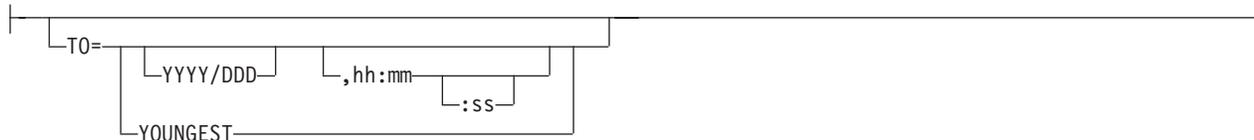


Figure 8. Log stream SUBSYS data set specification

**Note:** Quotation marks around SUBSYS-options1 and SUBSYS-options2 are required when non-alphanumeric characters are used in the options. They are not required when the option consists of a single keyword.

Other DD keywords will be validated, if specified, but will be ignored.

**DSNAME=log\_stream\_name**

specifies the name of the log stream to read. The name can be 1 to 26 characters in a data set name format.

**SUBSYS=(LOGR[exit\_routine\_name][,'SUBSYS-options1'][, 'SUBSYS-options2'])**

specifies that processing of this DD is to be handled by the **LOGR** subsystem.

**exit\_routine\_name**

The *exit\_routine\_name* is the second positional parameter and specifies the

name of the exit routine to receive control from MVS system logger. For log streams written by CICS, the exit routine name should be specified as DFHLGCNV.

CICS provides support for log streams generated by multiple CICS systems (a typical example would be a forward recovery log stream). Such log streams can contain log records generated by different releases of CICS. In order to ensure downward compatibility for all possible types of CICS log records, make sure that the highest level of DFHLGCNV (and its associated module DFHGTCNV) is referenced by batch jobs run against the log streams. As DFHLGCNV and DFHGTCNV reside in the SDFHLINK library, the MVS linklist should reference the SDFHLINK library of the highest release of CICS on an MVS region so that the batch jobs always use the highest available version of DFHLGCNV and DFHGTCNV.

**Note:** Omitting the `exit_routine` parameter, in order to read log records in the form they are stored on the log stream, is not a supported programming interface for CICS.

### SUBSYS-options1

specifies options meaningful to all exit routines.

**FROM=**{([yyyy/ddd][,hh:mm[:ss]]) | OLDEST}

indicates the starting time of the first log stream block to be processed. The first block processed will be the one with a time stamp greater than or equal to the specified time.

#### OLDEST

indicates that the first block read will be the oldest block on the log stream. If you omit the `FROM=` keyword, `OLDEST` is taken as the default.

*yyyy/ddd*

specifies the start date. If you omit a start date, the current date is assumed.

*yyyy* is a four-digit year number and *ddd* is a three-digit day number from 001 through 366 (366 is valid only on leap years). For example, code February 20, 2000 as 2000/051, and code December 31, 1996 as 1996/366.

*hh:mm[:ss]*

specifies the start time. If you omit the time, the first block written after midnight is used.

*hh* is a 2-digit hour number from 00 to 23, *mm* is a 2-digit minute number from 00 to 59, and *ss* is a 2-digit second number from 00 to 59. The seconds field and associated `:` delimiter can be omitted if not required by the log stream owner.

The `FROM=` keyword is mutually exclusive with the `DURATION=` keyword, and is not allowed when the `DELETE` keyword is specified.

**Note:** The time is GMT or local time, as selected in the `GMT|LOCAL` keyword.

**TO=**{([yyyy/ddd][,hh:mm[:ss]]) | YOUNGEST}

indicates the ending time of the last log stream block to be processed. The last block will be the one with a time stamp less than or equal to the specified time.

## YOUNGEST

indicates the last block read will be the youngest block on the log stream at the time the allocation for the DD occurs. If the TO= keyword is not specified, YOUNGEST is the default.

*yyyy/ddd*

specifies the end date. If the date is omitted, the current date is assumed.

yyyy is a four-digit year number and ddd is a three-digit day number from 001 through 366 (366 is valid only on leap years). For example, code March 7, 2001 as 2001/066, and code November 12, 2000 as 2000/317.

*hh:mm[:ss]*

specifies the end time. If the time is omitted, the last block written before midnight is used. If the end date is the same as the current day, the system uses the youngest block on the log stream at the time the allocation for the DD occurs.

hh is a 2-digit hour number from 00 to 23, mm is a 2-digit minute number from 00 to 59, and ss is a 2-digit second number from 00 to 59. The seconds field and associated : delimiter can be omitted if not required by the log stream owner.

The TO= keyword is mutually exclusive with the DURATION= keyword.

**Note:** The direction of the log stream browse is from the oldest (FROM=) to the youngest (TO=). If the value specified for the FROM= is greater than the value specified for the TO=, the jobstep is terminated with a JCL error.

The time is GMT or local time, as selected in the GMT|LOCAL keyword.

## 'DURATION=(nnnn,HOURS)'

The DURATION keyword is another method of requesting which blocks are to be processed. Each "n" is a numeric from 0 to 9. (nnnn,HOURS) requests the blocks for the "last nnnn hours" up to the youngest block to be processed. The "last nnnn hours" are calculated from the current time of the allocation for the DD.

The first block will be the one with a time stamp greater than or equal to the calculated start time. The last block read will be the youngest block on the log stream at the time the allocation for the DD occurs.

The DURATION= keyword is mutually exclusive with the TO= and the FROM= keywords.

#

## LOCAL|GMT

#

specifies whether the time is local time (based on the time zone offset at the time the log was written) or GMT time. LOCAL is the default.

## SUBSYS-options2

specifies exit routine unique options.

The options that are valid for CICS log streams when using exit routine DFHLGCNV are:

## LASTRUN

indicates that the starting point of the records to be read from the log stream is from the last record read by a previous use of a batch program that used LASTRUN. The end point of the records is to the youngest block in the log stream.

LASTRUN is mutually exclusive with the keywords in SUBSYS-options1.

**Note:** Only one last run point is associated with a log stream. You cannot, for example, specify LASTRUN on a daily log stream processing job and on a job run on a weekly basis

Note also that LASTRUN is not recommended when copying and deleting logstream data. See "Example 2" on page 69 for the recommended way of doing this.

## DELETE

indicates that log stream records are to be deleted from the log stream. The log stream itself is not deleted and remains available for use.

If the log stream has been opened in the job step, all records up to and including the last complete block read by the program are deleted from the log stream.

If the log stream has not been opened in the job step, all records prior to the TO= time are deleted from the log stream.

**Note:** This method of deleting records on an unopened log stream is much faster than deleting records on a log stream that has been opened.

If it is important that unprocessed records are not deleted, put the DELETE keyword as part of a conditional job step; see "Example 2" on page 69.

## COMPAT41 and COMPAT41V

specify the COMPAT41V option if applications are to run against the logstream and require the records to be presented, as far as possible, in the format used by CICS/ESA R4.1, rather than the format used by CICS Transaction Server for OS/390 Release 1.

Alternatively, DFHJUP can be run, using the COMPAT41 option and the NEWDCB option, to create a new dataset with the correct DCB information and the records, as far as possible, in the format used by CICS/ESA R4.1.

Batch applications requiring the records in COMPAT41 format can then be run against this new dataset.

Note that these batch applications do not need the COMPAT41 or COMPAT41V option to be specified.

---

## Using DFHJUP to read log streams

DFHJUP processes CICS journal data in MVS system logger log streams. It can also process journal data in SMF data sets. You may use multiple input log streams or SMF data sets, and format the output for multiple output data sets.

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.

Although the CICS log manager supports a maximum user data length of 62K bytes, the maximum record length readable through DFHJUP is 32K bytes. Data beyond the 32K-byte limit is not read and records are truncated at this point. 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 returns to the first stage, where the next group of tests is constructed.

The journal utility program runs as a standard operating system job. You can provide your own batch job to perform the function of DFHJUP. You must define a JOB statement, an EXEC statement, and DD statements defining input and output. "Examples of using DFHJUP" on page 68 gives some sample jobs that illustrate the use of DFHJUP.

## The DD statements

### STEPLIB DD

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

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

defines the input control data set. This file must be in 80-byte record format.

### Input or data DD for log stream processing

defines the input log streams to be examined to produce the output data.

The MVS image in which DFHJUP runs must be a member of the same sysplex as the MVS image in which the log stream was created. It is not necessary for the CICS region(s) that created the log stream, or any CICS region, to be running in the same MVS image as DFHJUP.

The default ddname is SYSUT1. The SUBSYS=(LOGR,DFHLGCNV,...) keyword identifies the DD statement as referring to a CICS log stream. You must specify SUBSYS keyword in any program that you use to examine and manipulate data in log streams; DFHJUP is the supplied program but you can use your own program to perform equivalent functions. See “SUBSYS=(LOGR,DFHLGCNV,...) keyword” on page 53 for information about the SUBSYS keyword.

The DCB parameter BLKSIZE=32760 must be specified on the input data DD statement if you are processing journal records on a log stream.

### **Input or data DD for SMF data set processing**

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.

**Note:** For CICS SMF 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 DSN=CICSLLOG,DISP=(OLD,KEEP),  
// DCB=RECFM=VB
```

The second example shows the use of the BLKSIZE parameter:

```
//SYSUT1 DD DSN=CICSLLOG,DISP=(OLD,KEEP),  
// DCB=RECFM=VB,BLKSIZE=32760
```

### **Output or data DD**

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 58.

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 2) 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 do not specify this statement, 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 2. 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 3) 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.

*Table 3. The DFHJUP OPTION statement*

1	10	16
OPTION	{PRINT   COPY   NEGOF}	[{OFFSET 0}={1 number}]
		[,{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}]
		[,NEWDCB]

Table 3. The DFHJUP OPTION statement (continued)

## 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 PRSYS 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 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.

**NEWDCB**

When using the COPY function, the DCB information from the original dataset is ignored when NEWDCB is specified. Instead the DCB information must be supplied on the JCL for the output dataset. This is especially useful in creating an output dataset in COMPAT41 mode from a logstream that is in the format used by CICS Transaction Server Release 1.

**END statement**

When you have defined all tests for the current input file, use END statement (see Table 4 on page 65) to initiate the tests.

Positions 10 and upward can be used for comments.

Table 4. The DFHJUP END statement

1	10	16
END	[.....comments.....]	

### COMMENTS statement

The COMMENTS statement (see Table 5) is optional. If used, it causes the contents to be displayed on the SYSPRINT data set.

Table 5. The COMMENTS statement

1	10	16
*		

## DFHJUP return codes

The following errors can occur in DFHJUP:

### Return code

#### Error description

- 04 This return code can be issued for any of the following reasons:
- 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.

## Managing the size of log streams

This section describes how to manage the size of logs.

### System log

*Normally, you should allow the CICS log manager to manage the size of the system log. You should not need to take explicit action to delete redundant data, nor to retain data—all system log data required on a restart is presented, providing the necessary completed unit of work information. If you *do* need to retain system log data beyond the time it would normally be deleted by CICS, see the *CICS Transaction Server for OS/390 Installation Guide* for advice on how to define your system log.*

## General logs

### Pre-OS/390 Release 3

In versions of MVS before OS/390 Release 3:

- The MVS system logger imposes a limit of 168 data sets per log stream.
- There is no mechanism for the automatic deletion of records from general log streams. It is your responsibility to delete such data to prevent the 168 data set limit being exceeded.

If you need longer-term data retention, then you must copy the data from log stream storage into alternative archive storage. See “Example 3” on page 70 for an example of the JCL you would need in a job to copy log stream data to archive storage, and then delete it from the log stream.

Although message IXG257I is issued when 90% of the log stream has been filled, this event is not detectable by CICS. You should use your automation software to monitor occurrences of this message.

### OS/390 Release 3 and later

In OS/390 Release 3 and later:

- The number of data sets per log stream recognized by the MVS logger is several million. In normal circumstances, you do not need to be concerned about the limit being reached.
- You can cause data to be retained on a log stream for a specified period, and then deleted automatically. To arrange this for general log streams, define the logs to MVS with AUTODELETE(YES) and RETPD(dddd), where dddd is the number of days for which data is to be retained. This causes the MVS system logger to delete an entire log data set when all the data in it is older than the retention period (RETPD) specified for the log stream.

**Note:** Support for the removal of the 168 data set limit, and for the AUTODELETE and RETPD parameters, requires the sysplex’s LOGR couple data set to have been formatted using OS/390 Release 3 or later. The removal of the 168 data set limit also requires the LOGR data set to have been formatted with DSEXTENT(nnnnn). If either has not been done, refer to the “Pre-OS/390 Release 3” box.

## Log data accessible to DFHJUP

DFHJUP is able to read both active and inactive data on the log stream. *Active data* is data that has not been deleted via an MVS IXGDELET request. *Inactive data* is data that has been deleted via an IXGDELET request, but which has not yet been physically deleted by MVS because of the retention period specified for the log stream.

As mentioned in Managing the size of log streams, if you are running under OS/390 Release 3 or later you can use the MVS RETPD parameter to specify a retention period for a log stream. If you specify a RETPD value greater than zero, MVS physically deletes data from the log stream only when both the following conditions are met:

1. The data is older than the retention period.
2. Either of the following applies:

- The data has been marked for deletion by an application (such as CICS or a utility program) issuing an IXGDELET request.
- AUTODELETE(YES) is specified for the log stream.

For definitive information about using the RETPD and AUTODELETE MVS parameters to automate the log tail deletion process, see the *CICS Transaction Server for OS/390 Installation Guide*.

### Example

Assume that you have defined a CICS system log with RETPD(10) and AUTODELETE(NO). The *active* portion of the log stream will consist of the data that CICS has not marked for deletion. The *inactive* portion of the log stream will consist of the data that CICS has marked for deletion, but which MVS has not yet physically deleted—because it is less than 10 days old.

Figure 9 shows active and inactive data on a log stream with a RETPD value of 10.

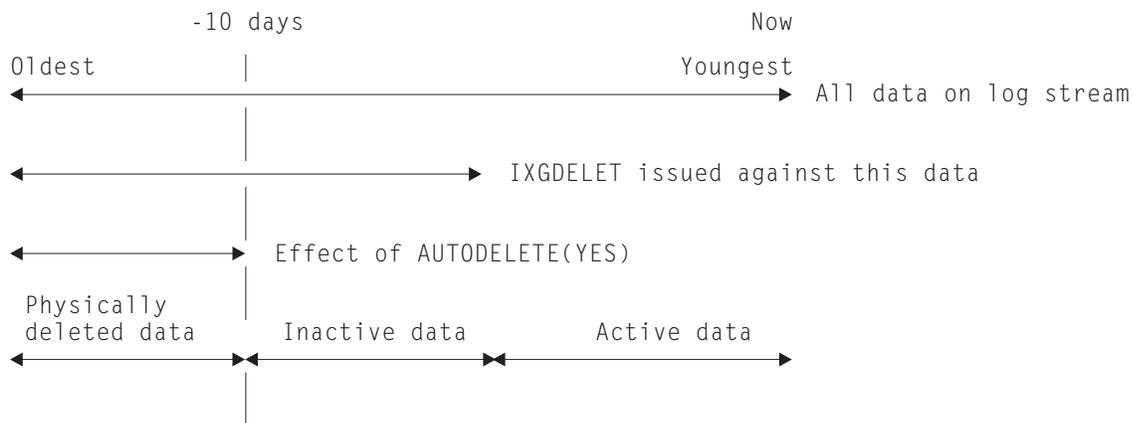


Figure 9. Active and inactive data on a log stream. The log stream has been defined with a RETPD value of 10.

The report output by DFHJUP advises you whether each block of data was read from the active or inactive area of the log stream—see Figure 10 on page 68.

## Diagnostic information in DFHJUP output

DFHJUP output provides diagnostic information for CICS system log streams, or CICS general log streams when COMPAT41 is not specified.

The block header record at the start of each log block is preceded by the following diagnostic information: MVS Block identifier, length of the block (in hexadecimal) and timestamps when the log block was written (in both GMT and local formats). The timestamps are displayed as both STCK values and formatted date and time fields. Note that the date field is in the format MM/DD/YYYY.

In addition, each log record in a block is preceded by a new column, which contains the offset (in hexadecimal) of the start of that log record from the start of the block.

See Figure 10 for an example of this diagnostic information.

```

Block identifier - 000000000008F2A
Length of block - 000000BD

GMT timestamp - AEEFF955B8400000 07/10/1997 13:47:36.980480
Local timestamp - AEF006BEF2800000 07/10/1997 14:47:36.980480

```

This block was read from the log stream active area

```

000000 000000 6EC4C6C8 00400001 C9E8C3D3 E9C3C3C3 AEEFE9CC 62CF0001 AEEFF721 96170001
000020 00000000 00000001
000000 000028 0000004C 00000038 00000014 AEEFF969 9E36E800 AEF006BE D17EE800 C3C5C3C9
000020 0000024C F8F7F3F6 0001D3C7 40404040 40404040 00000000 40F5F1F0 C9E8C3D3
000040 E9C3C3C3 E6D9C9C7 C8E3C140
000000 000074 00000049 00000038 00000011 AEEFF969 9ECA6000 AEF006BE D2126000 C3C5C3C9
000020 0000024C F8F7F3F6 0002E4D1 C4C6C8D1 F0F34040 00000000 0000000C C1E64040
000040 00000000 E3C5E2E3 F1

```

```

*>DFH. ..IYCLZCCC..Z.....7.0...*
*.....*
9E36E800 AEF006BE D17EE800 C3C5C3C9
*...<.....9...Y..0..J=Y.CECI*
40404040 00000000 40F5F1F0 C9E8C3D3
*...<8736..LG .... 510IYCL*
*ZCCCWRIGHTA *
9ECA6000 AEF006BE D2126000 C3C5C3C9
*.....9...-.0..K.-.CECI*
F0F34040 00000000 0000000C C1E64040
*...<8736..UJDFHJ03 .....AW *
*....TEST1 *

```

Figure 10. Diagnostic information in DFHJUP output

## Examples of using DFHJUP

The following examples illustrate some of the ways in which DFHJUP can be used. Each makes reference to a CICS log stream. However, this utility can be used with any data set that can be processed using QSAM.

**Note:** These examples refer to CICS general log streams, and NOT to the CICS primary or secondary system log streams DFHLOG or DFHSHUNT. CICS system log streams have different record formats and different field offsets within their log records.

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 11 on page 69 shows the JCL and control statements required to print to the output data set all the records written during a one-week period to a CICS general log.

```

//JNLPRNT1 JOB (accounting information),CLASS=A
//PRNTJNL EXEC PGM=DFHJUP
//STEPLIB DD DSNAME=CICSTS13.CICS.SDFHLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=A,DCB=RECFM=FBA
//SYSUT1 DD DSNAME=CICSDA#.CICSDA1.JRNL054,
//          DCB=BLKSIZE=32760,
//          SUBSYS=(LOGR,DFHLGCVN,
//          'FROM=(1995/001,06:00),TO=(1995/007,23:59),LOCAL')

```

Figure 11. DFHJUP program, example 1 (Part 1 of 2). JCL and control statements to print journal data on a CICS general log to an output data set

```

//SYSIN DD *
*-----*
* CONTROL STATEMENT : DEFAULTS *
* INPUT = SYSUT1 *
* OUTPUT = SYSPRINT *
* SELECTION QUALIFIERS : *
* 1. DEFAULT = ALL INPUT RECORDS *
*-----*
OPTION PRINT
END
*-----*
/*

```

Figure 11. DFHJUP program, example 1 (Part 2 of 2). JCL and control statements to print journal data on a CICS general log to an output data set

## Example 2

Figure 12 on page 70 shows the JCL and control statements required to copy to the output data set all the records written to a CICS general log. The records are copied in the CICS/ESA 4.1 format, and then deleted from the log stream. Note that there are two jobsteps, the first to perform the copy step, and the second to perform the delete step. The delete step is dependent upon the copy step completing successfully. Also, the same *TO=* value is specified on both jobstep's SUBSYS card. This ensures that both jobs manipulate the same range of log data. If *TO=* is omitted, and allowed to default to the current end of the logstream, there is the possibility of new log data being written to the logstream after the copy step, but before the delete step; this would then be deleted. To avoid this, it is recommended that the same *TO=* value be used for both jobsteps. Also, to avoid having to modify the *TO=* value every time the job is submitted, the year and Julian date are not specified. This causes the job to default to today's date. Therefore, the recommendation is to submit the job soon after the time specified on the *TO=* value. It should be noted that a *TO=* value close to midnight may cause problems. If the copy step were to run just before midnight, and the delete step just after midnight, the *TO=* value would default to different days, and there would be the potential for uncopied data being deleted.

```

//JNLCPY1 JOB (accounting information),CLASS=A
//COPYJNL EXEC PGM=DFHJUP
//STEPLIB DD DSNAME=CICSTS13.CICS.SDFHLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=A,DCB=RECFM=FBA
//SYSUT1 DD DSNAME=CICSAA#.CICSDC1.JRNL001,
//          DCB=BLKSIZE=32760,
//          SUBSYS=(LOGR,DFHLGCVN,'TO=(,10:00)',COMPAT41)
//SYSUT4 DD DSNAME=EXAMPLE1.COPY1,DISP=(NEW,CATLG),
//          UNIT=SYSDA,VOL=SER=USRPAAK,
//          SPACE=(TRK,(3,1))

```

Figure 12. DFHJUP program, example 2 (Part 1 of 2). JCL and control statements to copy journal data on a CICS general log to an output data set

```

//SYSIN DD *
*-----*
* CONTROL STATEMENT : DEFAULTS *
* INPUT = SYSUT1 *
* OUTPUT = SYSUT4 *
* SELECTION QUALIFIERS : *
* 1. DEFAULT = ALL INPUT RECORDS *
*-----*
OPTION COPY
END
//CHKCOPY IF (COPYJNL.RC = 0) THEN
//IEFBR14 EXEC PGM=IEFBR14
//LOGSTRM DD DSNAME=CICSAA#.CICSDC1.JRNL001,
//SUBSYS=(LOGR,DFHLGCVN,'TO=(,10:00)',,DELETE)
//SUBSYS=(LOGR,DFHLGCVN,')//CHKCOPY ENDIF
*-----*
/*

```

Figure 12. DFHJUP program, example 2 (Part 2 of 2). JCL and control statements to copy journal data on a CICS general log to an output data set

### Example 3

The next example shows how to delete a log-stream tail without reading the log stream.

```

//DELTAIL JOB (accounting information),CLASS=A
//IEFBR14 EXEC PGM=IEFBR14
//LOGSTRM DD DSNAME=CICSAA#.CICSDC1.JRNL001,
//          SUBSYS=(LOGR,DFHLGCVN,'TO=(1995/229,09:30)',DELETE)

```

Figure 13. IEFBR14 program, example 3. JCL and control statements to delete a log stream tail

### Examples of the use of the OPTION parameters

Depending on whether COMPAT41 has been specified on the SUBSYS parameter, log stream journal records are presented either:

- In the record format used at CICS/ESA 4.1, or
- In CICS Transaction Server for OS/390 Release 3 format (that is, the format introduced at CICS Transaction Server for OS/390 Release 1).

The OPTION parameters can be used to select specific types of records from a journal. You need to specify the offset within the record at which these specific record types lie. These offsets are different between the two different formats-CICS/ESA 4.1 and CICS Transaction Server for OS/390 Release 3.

See the *CICS Customization Guide* descriptions of the formats and offsets of fields in journal record headers.

There are tables at the end of this section to help you define the OPTION statements that you need. Example statements are included here to illustrate some of the types of record selection that can be achieved in this way.

### CICS Transaction Server for OS/390 Release 3 format

**Locating records using the system-type ID field:** If all the file control records were to be found, for example, the OPTION statement has the following form:

```
//SYSIN      DD *
OPTION PRINT OFFSET=43,FLDTYP=C,VALUE=FC,FLDLEN=2,COND=E
END
/*
```

The offset to this field, GLRH\_REC\_COMPID, is 39. If FLDTYP=C is used in the parameters, this value can be entered in its character form, as shown in the example above, for the component ID for file control, FC.

**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=34,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=29,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=29,FLDTYP=X,VALUE=E7F0F0F5,FLDLEN=4,COND=E
END
/*
```

**Finding all records with a particular time stamp:** At CICS Transaction Server for OS/390 Release 3, if you intend to select journal records for a particular time, you are recommended to use the time selection options on the SUBSYS parameter.

**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=37,FLDTYP=C,VALUE=T004,FLDLEN=4,COND=E
END
/*
```

**Selection using more than one search parameter:** Suppose you wanted to print all the file control records for a particular task. This needs two OPTION statements. The COND=M parameter performs the AND operation on the two statements.

```
//SYSIN      DD *
OPTION PRINT OFFSET=34,FLDTYP=X,VALUE=00025C,FLDLEN=3,COND=M
OPTION PRINT OFFSET=43,FLDTYP=C,VALUE=FC,FLDLEN=2,COND=E
END
/*
```

The example shows how to search for all records which belong to task number 25 and have a component ID of FC.

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 user journal records written with JTYPEID CP for transaction TRN5 are selected. The OPTION statements are 'ANDed' together.

```
//SYSIN      DD *
OPTION PRINT OFFSET=43,FLDTYP=C,VALUE=UJ,FLDLEN=2,COND=M
OPTION PRINT OFFSET=61,FLDTYP=C,VALUE=CP,FLDLEN=2,COND=M
OPTION PRINT OFFSET=29,FLDTYP=C,VALUE=TRN5,FLDLEN=4,COND=E
END
/*
```

## COMPAT41 format

**Locating records using the system-type ID field:** If all the file control records were to be found, for example, the OPTION statement has the following form:

```
//SYSIN      DD *
OPTION PRINT OFFSET=6,FLDTYP=X,VALUE=11,FLDLEN=1,COND=E
END
/*
```

The offset to this field, the module identifier, is 6. It is a numeric (X) type of field, of length 1 byte. For file control, this value equates to X'11' as listed in the *CICS Customization Guide*.

**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:** The time must be entered in the form hhhmss+ 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 file control records for a particular task. This needs two OPTION statements. 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=6,FLDTYP=X,VALUE=11,FLDLEN=1,COND=E
END
/*
```

The example shows how to search for all records which belong to task number 25 and have a system type ID of X'11'.

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=M
OPTION COPY OFFSET=23,FLDTYP=C,VALUE=TRN6,FLDLEN=4,COND=E
END
/*
```

## OPTION parameter values

Table 6. OPTION parameter values for CICS Transaction Server for OS/390 Release 3 journal records

Field name	OFF SET	FLD TYP	VALUE (example)	FLD LEN	Contents
<b>Note: General log header fields</b>					
GLRH_RECORD_LENGTH	1	X	00000100	4	Length of record
GLRH_HEADER_LENGTH	5	X	0000003B	4	Length of header
GLRH_REC_DATA_LEN	9	X	0050	4	Record data length
GLRH_GMT	13	X		8	Time (GMT)
GLRH_LOCAL	21	X		8	Time (local)
GLRH_TRAN_ID	29	C	TRN1	4	Transaction identifier
	29	X	E3D9D5F1	4	alternative format
GLRH_TASK_ID	33	X	0000025C	4	Task Number
GLRH_TERM_ID	37	C	T004	4	Terminal identifier
	37	X	E3F0F0F4	4	alternative format
GLRH_REC_TYPE	41	X	0001	2	Record type
GLRH_REC_COMPID	43	C	FC	2	Component ID
GLRH_REC_JOURNAL	45	C	JRNL0001	8	Journal name
	53	X	81	1	Start of task/start of UOW

Table 7. OPTION parameter values relevant for records presented in CICS/ESA 4.1 format

Field name	OFFSET	FLDTYP	VALUE (example)	FLDLEN	Contents
<b>Note: System header fields</b>					
JCROLL	1	X	0037	2	Length of record
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
<b>Note: Main system prefix fields</b>					
JCSPLL	11	X	0014	2	Length of system prefix
JCSPTASK	16	X	00025C	3	Task number
JCSPTIME	19	X	1445123F	4	Time of request - hhmmss+

Table 7. *OPTION* parameter values relevant for records presented in CICS/ESA 4.1 format (continued)

Field name	OFFSET	FLDTYP	VALUE (example)	FLDLEN	Contents
JCSPTRAN	23	C	TRN1	4	Trans-action identifier
	23	X	E3D9D5F1	4	alter-native format
JCSPTERM	27	C	T004	4	Terminal identifier
	27	X	E3F0F0F4	4	alter-native format



---

## Chapter 8. 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. 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 *OS/390 MVS System Management Facilities (SMF)*. 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 Performance Guide*. For a description of the STATRCD system initialization parameter, see the *CICS 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 Transaction Server for OS/390 Release 3 version of the DFHSTUP program, which you should use for CICS Transaction Server for OS/390 Release 3 data *only*.

---

### 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.

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 so access the relevant statistics record. You may 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 Supplied Transactions* manual. For programming information about the equivalent EXEC CICS commands, see the *CICS 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” 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 14 on page 79 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

```

Figure 14. Example job to extract and print statistics data (Part 1 of 6)

```

/*
//*****
/* 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=CICSTS13.CICS.SDFHLOAD,DISP=SHR
//SYSUT1   DD DSN=user.SMF.DATA,DISP=SHR
//SYSUT4   DD DSN=&&SMFSTATS,DISP=(NEW,PASS),UNIT=SYSDA,
//          SPACE=(CYL,(30,10))

```

Figure 14. Example job to extract and print statistics data (Part 2 of 6)

```

//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
/*

```

Figure 14. Example job to extract and print statistics data (Part 3 of 6)

```

//*****
/* Step 3: Sort, format and print the statistics records           4
//*****
//STUP1 EXEC PGM=DFHSTUP,REGION=32M
//*****
//STEPLIB DD DSN=CICSTS13.CICS.SDFHLOAD,DISP=SHR
//          DD DSN=CICSTS13.CICS.SDFHAUTH,DISP=SHR

```

Figure 14. Example job to extract and print statistics data (Part 4 of 6)

```

//DFHSTATS DD DSN=&&SMFSTATS,DISP=(OLD,DELETE)           5
//*DFHSTATS DD DSN=user.SMF.DATA,DISP=SHR
//DFHSTWRK DD UNIT=SYSDA,SPACE=(CYL,(8,4))              6
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(4))                7
//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))

```

Figure 14. Example job to extract and print statistics data (Part 5 of 6)

```

Notes:
//DFHPRINT DD SYSOUT=*                                   8
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SYSIN DD *                                           9
SELECT APPLID=(applid1,applid2)
COLLECTION TYPE=ALL
/*
//

```

Figure 14. Example job to extract and print statistics data (Part 6 of 6)

**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 14 on page 79, 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 *OS/390 MVS System Management Facilities (SMF)* .

**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 Figure 26 on page 123.)

**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). Unloading all the CICS SMF 110 records in this way also includes the SMF 110 records from journaling, monitoring, and the CICS Servers (temporary storage, coupling facility data tables and named counter). The DFHSTUP program only process CICS SMF 110 records with record subtype 2, all other SMF 110 records are ignored. If you want to unload just these CICS statistics records processed by DFHSTUP, change the record type parameter to TYPE(110(2)).

**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 61.

**4** The DFHSTUP program sorts statistics records in the sequence: specific applid, date (in DDYY form), and time.

**5** 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
```

**6** The ddname for the DFHSTUP work file must be DFHSTWRK. There are six types of statistics records that can be written to the DFHSTWRK data set:

- Files
- Log streams
- Transactions
- Transient data queues
- DB2 Entries
- TCP/IP Services

The size of the DFHSTWRK data set required will depend on the largest set of resources, from the above list, being written to the data set.

The following calculation can be used to estimate the size of the DFHSTWRK data set required:

- Files
  1. The length of the file statistics dsect, DFHA17DS, is 328 bytes.
  2. The amount of file data written in *one* interval/requested reset/End-of-day report is 312 \* (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 (328 bytes), should be added.
- Log streams
    1. The length of the log streams statistics dsect, DFHLGSDS, is 124 bytes.
    2. The amount of log stream data written in *one* interval/requested reset/End-of-day report is 124 \* (number of log streams in region) bytes.
  - Transactions
    1. The length of the transaction statistics dsect, DFHXRDS, is 104 bytes.
    2. The amount of transaction data written in *one* interval/requested reset/End-of-day report is 104 \* (number of transactions in region) bytes.
  - Transient data queues
    1. The length of the transient data queue statistics dsect, DFHTQRDS, is 136 bytes.
    2. The amount of transient data queue data written in *one* interval/requested reset/End-of-day report is 128 \* (number of transient data queues in region) bytes.
    3. Additionally, for extrapartition transient data queues, a record is written when the transient data queue is closed.
  - DB2 Entries
    1. The length of the DB2 entries dsect, DFHD2RDS is 156 bytes.
    2. The amount of DB2 entry data written in one interval/requested reset/End-of-day report is 156 \* (number of DB2 entries in region) bytes.
    3. Additionally, a record for each DB2 entry is written when the DB2 connection is closed.
  - TCP/IP Services
    1. The length of the TCP/IP services dsect, DFHSORDS is 108 bytes.
    2. The amount of TCP/IP service data written in one interval/requested reset/End-of-day report is 108 \* (number of TCP/IP services in region) bytes.
    3. Additionally, as TCP/IP service statistics are written to SMF whenever a TCP/IP service is closed online, an estimate of the number of TCP/IP services closes multiplied by the length of the DFHSORDS dsect (108 bytes), should be added.

**7** 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.

**8** The ddname for the output from the DFHSTUP program must be DFHPRINT, which you can direct either to a data set or printer.

**9** 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])}**

specifies 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 120 APPLIDs. If you specify more than 120 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])}**

specifies 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 120 APPLIDs. If you specify two or more APPLIDs, you must enclose them in parentheses, and separate them by commas. If you specify more than 120 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.

**SELECT TYPE={type | (type1[,type2]...[,typeN])}**

specifies the resource types 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. If you specify two or more resource types, you must enclose them in parentheses, and separate them by commas.

Code either the SELECT TYPE parameter or the IGNORE TYPE parameter but not both.

If you do not code this parameter, the DFHSTUP program reports statistics for all resource types found in the DFHSTATS data set, other than those resource types specified on an IGNORE TYPE parameter.

**IGNORE TYPE={type | (type1,type2)...[,typeN]}**

specifies the resource types for which you want the statistics ignored. The parameter keywords must be coded as shown, with one blank between the two keywords. If you specify two or more resource types, you must enclose them in parentheses, and separate them by commas.

Code either the SELECT TYPE parameter or the IGNORE TYPE parameter but not both.

The resource types that you can code on this parameter are:

- AUTOINSTALL
- CONNECTION
- DBCTL
- DB2
- DISPATCHER
- ENQUEUE
- FEPI
- FILE
- JOURNAL
- LOGSTREAM
- LSRPOOL
- MONITOR
- PROGAUTO
- PROGRAM
- RECOVERY
- STATS
- STORAGE
- SYSDUMP
- TCPIPSERV
- TABLEMGR
- TDQUEUE
- TERMINAL
- TRANCLASS or TCLASS
- TRANDUMP
- TRANSACTION
- TSQUEUE
- USER
- VTAM

If you do not code this parameter, the DFHSTUP program reports statistics for the resource types found in the DFHSTATS data set, depending on the SELECT TYPE parameter.

**COLLECTION TYPE={ALL | [,INT][,EOD][,REQ][,RRT][,USS]}**

specifies the statistics records to be included in the formatted reports for the selected APPLIDs, according to their collection type. 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, 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.

- DB2** Whenever CICS disconnects from DB2, CICS collects the statistics for the DB2 connection and all DB2ENTRYs covering the period from the last interval.

Whenever a DB2ENTRY is discarded, CICS collects the statistics for that DB2ENTRY covering the period from the last interval.

#### **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.

#### **Journalnames**

Unsolicited Journalnames statistics are produced when a journalname is discarded (DISCARD JOURNALNAME).

### **Logstreams**

Unsolicited Logstream statistics are produced when a logstream is disconnected from the MVS logger.

### **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.

### **TCP/IP services**

Unsolicited TCP/IP service statistics are produced whenever a TCP/IP service is closed.

### **System dumps**

Whenever a system dump table entry is deleted, CICS collects the statistics covering the 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.

### **Transaction dumps**

Whenever a transaction dump table entry is deleted, CICS collects the statistics covering the period since the last interval.

### **Transient data queues**

Unsolicited Transient data queue statistics are produced when a transient data queue is discarded (DISCARD TDQUEUE), or when an extrapartition transient data queue is closed.

### **TIME START=hh.mm.ss,STOP=hh.mm.ss,ELAPSED|DAILY**

specifies that the DFHSTUP program is to print only the statistics collected during the specified period. The period is determined by a combination of the START time to STOP time, the ELAPSED|DAILY keyword and the DATE parameter as follows:

#### **ELAPSED**

This is the default. If ELAPSED is coded, DFHSTUP will process every statistics record between the DATE START and TIME START until the DATE STOP and TIME STOP.

## DAILY

If DAILY is coded, DFHSTUP will process every statistics record between the TIME START and TIME STOP for each day between the specified DATE START and DATE STOP.

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.

## Examples

1. To process every statistics record written between 3rd June 1997 at 10:00 hours and 10th June 1997 at 20:00 hours, you can code the following TIME and DATE control statements:

```
TIME START=10.00.00,STOP=20.00.00,ELAPSED
```

```
DATE START=06/03/97,STOP=06/10/97
```

2. To process every statistics record written between 10:00 hours and 20:00 hours each day starting on 3rd June 1997 and stopping on 6th June 1997, you can code the following TIME and DATE control statements:

```
TIME START=10.00.00,STOP=20.00.00,DAILY
```

```
DATE START=06/03/97,STOP=06/10/97
```

### DATE START=mm/dd/yy or mm/dd/yyyy,STOP=mm/dd/yy or mm/dd/yyyy

specifies that the DFHSTUP program is 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

For the twenty-first century the year must be represented by yyyy. If yy is coded the twentieth century is assumed. For example, a date of 12/20/96 is for the 20th December 1996 and a date of 12/20/2005 is for the 20th December 2005.

## PAGE SIZE=number

specifies the number of lines to be formatted per page, in the range 20 to 99. The default page size is 60 lines.

## SUMMARY

specifies that the DFHSTUP program is 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
- Recovery Manager
- Enqueue Manager
- 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
- Table manager
- Transaction
- Program
- File
- LSR Pool
- LSR Pool file
- Transient data queue
- Journalname
- Logstream
- DB2 connection and DB2ENTRYs
- TCP/IP services
- Terminal
- ISC/IRC system and mode entry
- ISC/IRC system security
- DBCTL
- FEPI pool
- FEPI connection
- FEPI target
- User domain.

**Note:** The statistics produced in the summary report for SELECT TYPE(LSRPOOL) do not contain buffer information for individual LSR pools.

If the SMF data set (or data sets) contains CICS statistics from several runs of CICS with the same applid, you must use the TIME parameter, and if necessary the DATE parameter, to produce the summary report for one run of

CICS. If you do not use the TIME and DATE parameters to specify one of several runs of CICS, the results are unpredictable.

You can save a lot of paper if you code this parameter and omit the COLLECTION TYPE parameter.

If this parameter is not coded, a summary report is not produced.

**UPPERCASE=YES**

specifies that the statistics output is to be in uppercase only. The parameter must be coded as shown in uppercase characters with no spaces between words. The parameter must be the first one coded in the parameter list. If you want output in mixed case (the default), do not code this parameter.



#

---

## # Chapter 9. Recovery statistics program (DFH\$STER)

# The recovery statistics program, DFH\$STER, replaces the statistics produced by the  
# obsolete recovery utility, DFHRUP. DFH\$STER retrieves global statistics from the  
# Recovery Manager, formats and prints the information to the CICS transient data  
# destination, CSSL, as used by the recovery utility program (DFHRUP) for writing  
# statistics. DFH\$STER enhances the Recovery Manager Domain recovery messages  
# produced on a CICS emergency restart.

#

---

### # Recovery Manager domain recovery messages

# During an emergency restart, the Recovery Manager produces several messages  
# detailing the state of inflight and shunted units of work (UOWs). The messages are  
# produced by the Recovery Manager after processing the system log, and they  
# accurately represent the state of the CICS system workload at the time the  
# cancellation, abend, or termination occurred. When the messages are produced, the  
# Recovery Manager attempts to resolve the UOWs by committing, backing-out, or  
# temporarily suspending completion (shunting) them.

# The messages are as follows:-

#

#### DFHRM200

#

<applid> xx Indoubt UOWs were reconstructed

#

# After reading the system log, the CICS Recovery Manager found that there was  
# the specified number of UOWs that were shunted indoubt, or were waiting  
# indoubt at the time the system was terminated. Both types have suffered an  
# indoubt failure with their recovery coordinator, for example, another CICS, or  
# DBCTL. They must wait to be reconnected or resynchronized with the  
# coordinator before they can commit or backout the changes made within a  
# particular shunted UOW, that is to complete a syncpoint. Recovery  
# coordinators will be across LU6.2, LU6.1, MRO (XM/IRC/XCF) links, or a  
# product interface through the Resource Manager Interface (RMI) such as  
# DBCTL, MQ or DB2.

#

#### DFHRM201

#

<applid> xx backout-failed and commit-failed UOWs were reconstructed

#

# After reading the system log, the CICS Recovery Manager finds a specified  
# number of UOWs shunted while awaiting the availability of a LOCAL  
# recoverable resource that was updated by the task in question. The local  
# resources to CICS are files, RLS files, TD queues, TS Queues, RDO objects and  
# user. There are many reasons why a recoverable resource cannot be committed  
# or backed out, for example, RLS server not available, I/O errors on datasets,  
# coupling facility errors, backout exit failures. As with the message DFHRM200,  
# the UOWs could not complete their syncpoint processing. However, they are  
# not shunted while awaiting resolution of an indoubt failure.

#

#### DFHRM202

#

<applid> xx inflight UOWs were reconstructed

#

# After reading the system log, the CICS Recovery Manager finds a specified  
# number of UOWs in progress. That is, the number of tasks that were executing  
# in the CICS region before the system was terminated abnormally. These tasks

# are in between syncpoint requests and therefore have to be backed-out  
# (rolledback) or shunted awaiting resynchronization with an external resource  
# manager such as DBCTL.

# **Note:** 'xx' in each case represents the number of UOWs in the respective states.

---

# **Statistics sample program, DFH\$STER**

# The statistics sample program, DFH\$STER, shown in Figure 15, retrieves and  
# formats the recovery manager global statistics and writes the formatted output to  
# the CICS transient data destination, CSSL.

```
DFHRM0201 10/29/99 11:10:40 IYK2Z1V1 0 backout-failed and 1 commit-failed OUWs were reconstructed
DFH$STER : CICS Restart Recovery Activity report

No. of syncpoints (forward)           : 14
No. of syncpoints (backward)          : 0
No. of resynchronizations              : 0

No. of UOW indoubt failure shunts completed : 0
No. of UOW com/back failure shunts completed : 1

No. of UOW indoubt failure shunts outstanding : 0
No. of UOW com/back failure shunts outstanding : 0

DFH$STER : End of CICS Recovery activity report
```

*Figure 15. Statistics sample program, DFH\$STER*

# For further information about the shunting of UOWs, see the *CICS Recovery and*  
# *Restart Guide*.

---

# **Installation**

# This program should be defined in the 2nd phase PLTPI for your CICS region. See  
# the *CICS Resource Definition Guide* .

---

## Chapter 10. Trace utility print program (DFHTU530)

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 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, DFHTU530
- The GTF data sets, using a CICS-supplied routine with the MVS interactive problem control system (IPCS).

---

### The CICS trace utility program, DFHTU530

The CICS utility program, DFHTU530, 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=DFHTU530 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 numbers <sup>3</sup>
- 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.

If more than the specified interval elapses before the next trace entry is written, 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 16 on page 94 to invoke the utility program for each auxiliary trace data set. Only use the trace utility program

---

3. The sequence number is given in each trace entry, and can be determined from a summary trace point.

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,
//          REGION=2M
//PRINT  EXEC  PGM=DFHTU530
//STEPLIB DD  DSN=CICSTS13.CICS.SDFHLOAD,DISP=SHR
//DFHAUXT DD  DSN=CICSTS13.CICS.DFHxxx,DISP=SHR
3 //DFHAXPRT DD  SYSOUT=A
//DFHAXPRM DD  *
:
:
[trace selection parameters]
/*
```

Figure 16. 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 DFHTU530 utility. You can use the sample region size as a basis for your own JCL, but you must ensure that the region size is large enough to run the DFHTU530 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=CICSTS13.CICS.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=DFHTU530,PARM='selection_parameter,selection_parameter,...'
```

#### PAGESIZE=(value)

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 | SHORT | FULL**

specifies 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.

**SHORT**

indicates that you require the short formatted print of the data in each entry. This consists of the information in the abbreviated format entry, and the following elements from the interpretation string of the fully formatted entry:

- Interpreted parameter list
- Return address
- Time
- Interval

**FULL** indicates that you want a fully formatted print of all the data in each entry. This is the default.

**ALL**

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}),,,,]}

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**

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}

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 Transaction Server for OS/390 Release 3 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,.,.])**

specifies that only the entries for tasks with the specified hexadecimal kernel task numbers are printed.

**PAGESIZE=(value)**

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},,..])**

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,.,.])**

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 DFHTU530 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 97.

**TRANID=(tttt[,tttt,.,.])**

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 DFHTU530 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 97.

**TIMERG=(hhmmss-hhmmss[,hhmmss-hhmmss,.,.])**

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 DFHTU530 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 DFHTU530 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}])**

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:

<b>AP</b>	Application
<b>BR</b>	3270 bridge
<b>DD</b>	Directory manager
<b>DE</b>	DCE services
<b>DM</b>	Domain manager
<b>DS</b>	Dispatcher
<b>DU</b>	Dump
<b>EX</b>	External CICS interface
<b>GC</b>	Global catalog
<b>KE</b>	Kernel
<b>LC</b>	Local catalog
<b>LD</b>	Loader
<b>LG</b>	Log manager
<b>LM</b>	Lock manager
<b>ME</b>	Message
<b>MN</b>	Monitoring
<b>NQ</b>	Enqueue
<b>PA</b>	Parameter manager
<b>PG</b>	Program manager
<b>RM</b>	Recovery manager
<b>SM</b>	Storage manager
<b>ST</b>	Statistics
<b>TI</b>	Timer
<b>TR</b>	Trace
<b>US</b>	User
<b>XM</b>	Transaction manager
<b>XS</b>	Security manager

**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.

#### **UPPERCASE**

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 17 on page 99.

If you select trace entries by specifying the TRANID or TERMID parameters, the DFHTU530 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 17 on page 99, 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,wwww)

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
```

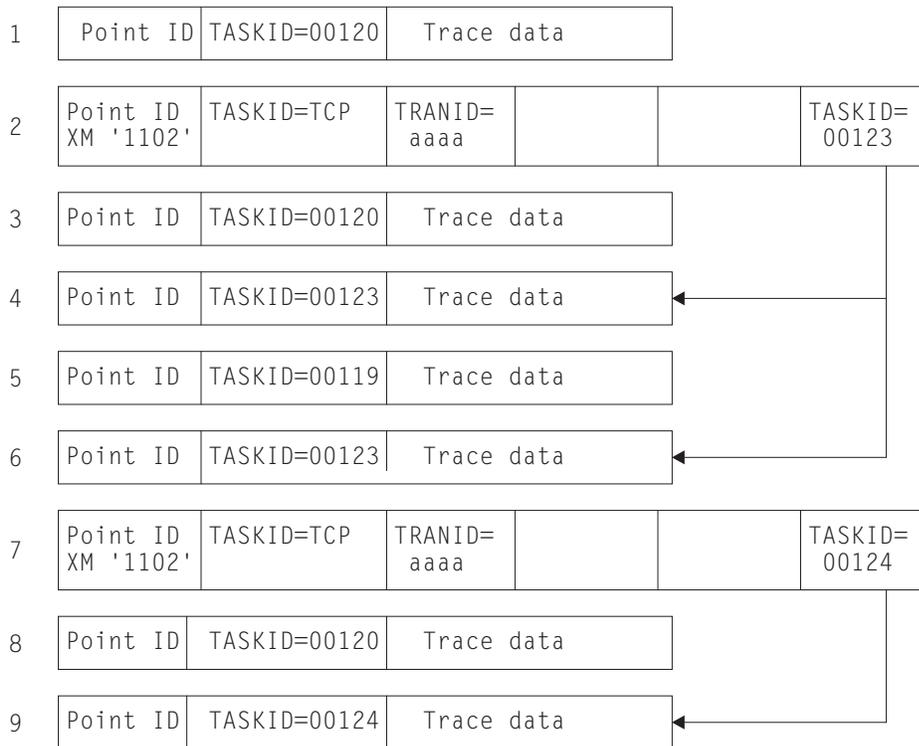
The same example could be coded in the PARM parameter as follows:

```
//      EXEC  PGM=DFHTU530,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
```

Start of auxiliary trace data:



# **Note:** The trace point for transactions that have been task-attached is XM 1102. The trace points for transactions that  
 # have been terminal-attached are AP 1730, AP 1731, AP 1790, AP 1791, AP 17B0, AP 17B1, AP 17C0 and AP  
 # 17C1.

Figure 17. 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 DFHTU530 selective trace control statements on the CICS(text) parameter. The CICS-supplied formatting routines are called DFHTG530 and DFHTR530, supplied in CICSTS13.CICS.SDFHLINK. DFHTG530 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 Transaction Server for OS/390 Release 3, CICS Transaction Server for OS/390 Release 2, CICS Transaction Server for OS/390 Release 1, CICS/ESA 4.1, and CICS/ESA 3.3 GTF trace entries written to GTF using the same GTFTRACE command. To do so, you must make available the formatting routine for CICS Transaction Server for OS/390 Release 3 (DFHTG530, its alias AMDUSREF, and DFHTR530), CICS Transaction Server for OS/390 Release 2, (DFHTG520, its alias AMDUSREF, and DFHTR520), CICS/ESA 4.1 (DFHTG410, its alias AMDUSREF, and DFHTR410), and the formatting routine for CICS/ESA 3.3, called DFHTRVR3. The DFHTRVR3 module is provided by APAR PN59159 for CICS330.SDFHLINK.

## 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 DFHTR530 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, DFHTR530. You can code any of the DFHTU530 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 18 on page 101. For further information about the PROFILE PAGESIZE subcommand, see the *OS/390 MVS IPCS Commands*. (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 *OS/390 MVS IPCS Commands*. (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 *OS/390 MVS IPCS Commands* .

### 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 18.

**Notes:** **1** The batch job in Figure 18 includes a STEPLIB statement for the load

```
//IPCSGTF JOB ( accounting info),
//          CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=userid
//PRINTTR EXEC PGM=IKJEFT01,REGION=4096K
//STEPLIB DD DSN=CICSTS13.CICS.SDFHLINK,DISP=SHR 1
//          DD DSN=CTS110.CICS510.SDFHLINK,DISP=SHR
//DFHTRACE DD DSN=GTF.TRACEnn,DISP=SHR
//IPCSDDIR DD DSN=ipcs.dump.directory,DISP=SHR 2
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//IPCSPRNT DD SYSOUT=*
//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 18. Sample IPCS job to print CICS trace entries from a GTF data set

libraries that contain the modules (DFHTG530, its alias AMDUSREF, DFHTR530, DFHTG530, its alias AMDUSREF and DFHTR530) 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:

STEPLIB	CICS releases to be formatted
CICSTS13.CICS.SDFHLINK	CICS Transaction Server for OS/390 Release 3 only.
CICSTS13.CICS.SDFHLINK CICSTS12.CICS.SDFHLINK	CICS Transaction Server for OS/390 Release 3, and CICS Transaction Server for OS/390 Release 2
CICSTS13.CICS.SDFHLINK CICSTS11.CICS.SDFHLINK	CICS Transaction Server for OS/390 Release 3, and CICS Transaction Server for OS/390 Release 1
CICSTS13.CICS.SDFHLINK CICS410.SDFHLINK	CICS Transaction Server for OS/390 Release 3, and CICS/ESA 4.1.
CICSTS13.CICS.SDFHLINK CICS330.SDFHLINK	CICS Transaction Server for OS/390 Release 3, and CICS/ESA 3.3.

**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 24 on page 116.

### **Invoking the CICS formatting routine from TSO**

To enable the CICS formatting routines to be invoked from TSO, copy the modules DFHTG530, AMDUSREF, DFHTR530, 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 11. Dump utility program (DFHDU530)

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. CICS provides two utility programs to help you analyze these dumps. These are:

1. The transaction dump utility program, DFHDU530, for processing CICS transaction dumps. For details of this CICS batch utility program, see “The CICS transaction dump utility program, DFHDU530”.
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 109.)

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 108.

---

### The CICS transaction dump utility program, DFHDU530

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, DFHDU530, prepares the transaction dump output for printing and prints the formatted information.

#### Selecting transaction dump output for the DFHDU530 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 DFHDU530 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 DFHTINDEX, or to the SYSPRINT data set if the DFHTINDEX 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 19. SYSIN control statements for the DFHDU530 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 SYSPRINT data set if the DFHTINDX DD statement is missing. If you code SCAN, all other statements in the SYSIN data set (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]...)**

specifies that dumps are to be selected by their transaction identifier (ID). You can code up to 20 4-character 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 fewer 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 3-character transaction ID of 'ABC'.

**TRANID=AB+**

specifies a 3-character transaction ID, where the first two are 'AB', and the third is any character (other than blank).

**TRANID=CD+F**

specifies a 4-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}],,,)**

specifies that dumps are to be selected by a transaction dump code, which is either the 4-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}],,,)**

specifies that dumps are to be selected 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 DFHDU530 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 DFHDU530 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=INITIAL or 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**

specifies a single dump identified as the fifth dump taken during dump run number 10.

**DUMPID=125/0001-125/9999**

specifies all the dumps taken during dump run number 125.

**DUMPID=(125/0001-125/0003,125/0019)**

specifies the first three dumps taken during dump run number 125, plus dump count number 19.

**PAGESIZE=(value)**

specifies the number of lines to be 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}],,,)**

specifies that dumps are to be selected 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**

specifies that the data output is to be 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 DFHDU530 program

The job stream to run the DFHDU530 program should include DD statements for the following data sets:

**DFHDMPPDS (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, DFHDU530 tries to open a SYSPRINT data set for dump index output. If it is 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 DFHDU530 program, needed if you want to code specific selection parameters. If you omit the SYSIN DD statement, you get message IEC130I.

See Figure 20 for a sample job stream for the DFHDU530 program.

```
//PRNTDMP JOB accounting info,name,MSGLEVEL=(1,1),
//          CLASS=A,MSGCLASS=A,REGION=2M
//          EXEC PGM=DFHDU530,PARM='command,command,...'
//STEPLIB DD DSN=CICSTS13.CICS.SDFHLOAD,DISP=SHR
//DFHMPDS DD DSN=CICSTS13.CICS.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
/*
//
```

**1**

Figure 20. Sample job to format and print CICS transaction dump data sets

### **Note:**

**1** The sample JCL gives a region size of 2MB that you might typically need to run the DFHDU530 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 DFHDU530 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 not erased, but they are lost 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.

If you use the dump utility program to print a dump data set that is still in use by CICS, 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.

Do not use the dump utility program to print a dump data set that has not been opened during the most recent execution of CICS. If you try to, either transaction dumps from a previous execution are reprinted, 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 dump 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 111.

For further information about IPCS, see the *OS/390 MVS IPCS User’s Guide*.

## Preparing to 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”.)
- Ensure that the DFHIPCSP member can be found by your IPCS job. (See “Specifying DFHIPCSP CICS exit control data needed”.)
- 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 Transaction Server for OS/390 Release 3” on page 110.)
- 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 110.)

### 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 CICSTS13.CICS.SDFHPARM library, contains the CICS release-specific entries for the IPCS exit control table. These entries are listed in Figure 21 on page 110.

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=CICSTS13.CICS.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 Transaction Server for OS/390 Release 3

In CICS Transaction Server for OS/390 Release 3, 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')
EXIT EP(DFHPD510) VERB(CICS510) ABSTRACT(+
  'CICS Transaction Server for OS/390 Release 1 analysis')
EXIT EP(DFHPD520) VERB(CICS520) ABSTRACT(+
  'CICS Transaction Server for OS/390 Release 2 analysis')
EXIT EP(DFHPD530) VERB(CICS530) ABSTRACT(+
  'CICS Transaction Server for OS/390 Release 3 analysis')
/* ===== */
```

Figure 21. Release-specific entries in DFHIPCSP for DFHPDnnn routines

### Modifying IPCS jobs on releases before CICS Transaction Server for OS/390 Release 3

IPCS jobs on CICS releases before CICS Transaction Server for OS/390 Release 3 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 21 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 Transaction Server for OS/390 Release 3, DFHPD530, is installed in the SYS1.CICSTS13.CICS.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, DFHPD530.

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 Transaction Server for OS/390 Release 3” on page 110.)
- Copy the routines to the SYS1.CICSTS13.CICS.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 the 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 CICS530 'DEF=1,DLI=1,KE=3,TR=2,TRS=<TRANID=CSSC,KE_NUM=12>'
```

### Notes:

1. The VERBEXIT statement specifies the verb name CICS530 to process CICS Transaction Server for OS/390 Release 3 system dump data. This corresponds to the IPCS dump exit routine DFHPD530, as specified in the DFHIPCSP member in the CICSTS13.CICS.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 94.

## 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 21 on page 110 for the CICS-supplied verb names.)

The syntax of the CICS exit parameters is shown in Figure 22 and described in “The CICS dump exit parameters” on page 112.

For some examples of using IPCS to process CICS SDUMPs, see page 113.

```
[JOB={jobname|CURRENT}]  
[UPPERCASE]  
  
[,DEF={0|1|2|3}]  
[,keyword [=levelnumber]]
```

Figure 22. The CICS dump exit parameters

where

**keyword**

specifies the CICS component ID (see page 113 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 113 for details of the available level numbers).

**The CICS dump exit parameters**

The CICS dump exit parameters are as follows:

**JOB={jobname | CURRENT} (optional)**

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 using 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 (optional)**

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}**

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 CICS530 '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 CICS530 '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 CICS530 '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 115.

#### Examples of the use of level numbers

VERBEXIT CICS530 'JOB=CURRENT,KE=1,DS' prints a summary of the kernel domain data, and all available information for the dispatcher domain.

VERBEXIT CICS530 '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 CICS530 without any parameters produces summary and control block output for all the CICS components in the dump.

#### The CICS530 dump exit component keywords

The component keywords specify which functional areas of the CICS dump you want the CICS530 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. 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 DFHTU530 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 DFHTU530 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.

For details about the CICS dump component keywords, see the *CICS Problem Determination Guide*.

The CICS530 dump exit can be used in either a batch job or interactively. For an example of a batch IPCS job, see Figure 25 on page 117. For information about using IPCS, see the following MVS IPCS manuals:

- *OS/390 MVS IPCS User's Guide*.
- *OS/390 MVS IPCS Commands*.

**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 DFHTU530 for the formatting of CICS auxiliary trace entries, except the parameters PAGESIZE, ABBREV, SHORT, and FULL. You may, as with DFHTU530, select any number of parameters from those available. (For descriptions of available parameters, see "The trace selection parameters" on page 94.)

**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 DFHTU530 to format auxiliary trace entries.

APAR 39051

# Documentation added 18/09/00

# **Application selection::** The APS component keyword will format application  
# domain and Language Environment storage areas for the specified task identifier  
# only.

# The application selection parameter specifies the task identifier of a single task for  
# which storage areas are to be printed. The task identifier value can be any number  
# up to five digits, or an alphanumeric value of three characters.

# **Note:** You must use angled brackets around the specified parameter.

## # **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 23 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 23. 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 24, is an example of how to create an IPCS dump directory; the second, in Figure 25 on page 117, 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 CICS530 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.
//*****
```

Figure 24. Sample job to create an IPCS dump directory (Part 1 of 4)

```
//CREDDIR EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//AMSDUMP DD SYSOUT=*
//SYSIN DD *
```

Figure 24. Sample job to create an IPCS dump directory (Part 2 of 4)

```
DEFINE CLUSTER (NAME(CICSTS13.CICS.IPCSDIR)          -          1
                VOLUMES(vol1d)                      -          2
                CYLINDERS(2 1)                       -
                BUFFERSPACE(65536)                   -
                KEYS(128 0))                          -
                DATA( NAME(CICSTS13.CICS.IPCSDIR.DATA) -          1
                CONTROLINTERVALSIZE(4096))           -
                INDEX(NAME(CICSTS13.CICS.IPCSDIR.INDEX)) -          1
```

Figure 24. Sample job to create an IPCS dump directory (Part 3 of 4)

```
/*
//INITDIR EXEC PGM=IKJEFT01,REGION=1500K
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
IPCSDIR 'CICSTS13.CICS.IPCSDIR'          1
END
/*
//
```

Figure 24. Sample job to create an IPCS dump directory (Part 4 of 4)

### Notes:

- 1** Change 'CICSTS13.CICS' 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 25 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=CICSTS13.CICS.SDFHLINK,DISP=SHR
//          DD DSN=CICSTS13.CICS.SDFHPARM,DISP=SHR
//DFHSDUMP DD DSN=DUMP.NAME,DISP=SHR
//DFHSNAP DD SYSOUT=*
//IPCSDDIR DD DSN=CICSTS13.CICS.IPCSDIR,DISP=SHR
//IPCSPARM DD DSN=SYS1.PARMLIB,DISP=SHR
//          DD DSN=CICSTS13.CICS.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 CICS530 'UPPERCASE,DEF=1,DLI=2,KE=3,TR=2'
END
/*
```

Figure 25. Sample job to format a CICS SDUMP using IPCS and the CICS dump exit

**Notes:**

- 1** The batch job in Figure 25 includes STEPLIB statements for:
  - The CICSTS13.CICS.SDFHLINK library, to enable the VERBEXIT subcommand of IPCS to invoke the CICS-supplied IPCS dump exit routine DFHPD530
  - The CICSTS13.CICS.SDFHPARM library, to enable the IPCS job to find the CICS-supplied DFHIPCSP member.

**2** Specify the name of the dump data set being processed instead of 'DUMP.NAME'.

**3** The DD statement for DFHSNAP 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 DFHSNAP. If you omit the DD statement, you get message IEC130I. We recommend that you always include the DD statement.

**4** Change 'CICSTS13.CICS' 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 "Chapter 2. Starting up CICS regions" on page 17.

**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, 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.SBLCSLI0 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 CICS530 to process CICS Transaction Server for OS/390 Release 3 system dump data. This corresponds to the IPCS dump exit routine DFHPD530, as specified in the DFHIPCSP member in the CICSTS13.CICS.SDFHPARM library.

---

## Chapter 12. Monitoring utility programs (DFHMNDUP and DFH\$MOLS)

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 Customization Guide*. For information about the MCT definitions, see the *CICS Resource Definition Guide*.

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 8.

Table 8. Relationship of DFHMNDUP SYSIN parameters to SMF fields

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=yyyydd**

specifies the Julian date to be included in the dictionary record, where:

- yy** represents the year of the twentieth century (for example 98 for 1998).
- 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 96354 represents the 20th December 1996 and the date 2005354 represents the 20th December 2005. If you do not specify a date, the current date is used.

**GAPPLID=name**

specifies the generic VTAM APPLID of the CICS region for which you are analyzing performance data.

# **Note:** It is important not to confuse the term “generic applid” with “generic resource name”.

# Remember that “generic” and “specific” applids apply only to systems that use XRF. CICS systems that do not use XRF have only one applid, which is the value defined in the “APPLID=3D” parameter.

- # • For XRF, a CICS system’s **generic applid** is defined on the APPLID system initialization parameter and is the name by which CICS is
- #

# known in the network. (That is, it is the name quoted by remote CICS  
 # systems, on the NETNAME option of CONNECTION definitions, to  
 # identify this CICS.)

- # • A CICS system's **specific applid** is used to distinguish between the  
 # pair of XRF systems. It is the name quoted on a VTAM APPL  
 # statement, to identify this CICS to VTAM.
- # • A CICS system's **generic resource name** is defined on the GRNAME  
 # system initialization parameter, and enables CICS to become a  
 # member of a VTAM generic resource group.

# Note, in particular, that you cannot use both VTAM generic resources  
 # and XRF. If you use VTAM generic resources, you should specify only  
 # one name on the APPLID system initialization parameter.

**JOBDATE=yyddd or JOBDATE=yyyyddd**

specifies 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, 98 for 1998).

**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 96354 represents the 20th December 1996 and the date 2005354 represents the 20th December 2005. If you do not specify a date, the current date is used.

**JOBNAME=xxxxxxx**

specifies an MVS job name for the CICS region to be included in the dictionary record.

**JOBTIME=hhmmss**

specifies 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**

specifies 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=,'.

# **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**

specifies 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**

specifies the system identifier of the MVS system that owns the SMF data sets.

**TIME=hhmmss**

specifies a time stamp for the dictionary record. If you do not specify a time, the current time is used.

**UPPERCASE**

specifies that you want the statistics output in uppercase only. If you want output in mixed case (the default), do not code this parameter.

**USERID=xxxxxxxx**

specifies 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 *OS/390 MVS Installation Exits* manual, SC28-1753.

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 26 on page 123 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=CICSTS13.CICS.SDFHLOAD,DISP=SHR           1
//          DD DSN=mct.table.loadlib,DISP=SHR
//SYSUT4  DD DSN=CICSTS13.CICS.applid.MNDUPREC,DISP=(NEW,CATLG),
//          UNIT=SYSDA,SPACE=(TRK,(1,1))                 2
//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')    3
//OUTDD1  DD DSN=SYS1.SMFDMPn,DISP=(NEW,CATLG),           4
//          SPACE=(CYL,(10,2)),UNIT=SYSDA
//SYSPRINT DD SYSOUT=A
//SYSIN   DD *
          INDD(INDD1,OPTIONS(DUMP))
          OUTDD(OUTDD1,TYPE(110(1)))
//*****
//*   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=CICSTS13.CICS.SDFHLOAD,DISP=SHR
//INPUT   DD DSN=CICSTS13.CICS.applid.MNDUPREC,DISP=OLD   5
//          DD DSN=SYS1.SMFDMPn,DISP=OLD
//SYSPRINT DD SYSOUT=A
//SYSABEND DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//SYSIN   DD *
(DFH$MOLS control statements - see page 127)

```

Figure 26. 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 *OS/390 MVS System Management Facilities (SMF)* 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 *OS/390 MVS System Management Facilities (SMF)* 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.

The DFH\$MOLS program is enhanced as follows for CICS Transaction Server for OS/390 Release 3:

- It can handle SMF 110 monitoring data records for CICS Transaction Server for OS/390 Release 3, CICS Transaction Server for OS/390 Release 2, and CICS Transaction Server for OS/390 Release 1, in addition to CICS/ESA Version 3 and CICS/ESA Version 4.
- It can unload performance class records into a flat file (CICS Transaction Server for OS/390 Release 3 performance class records only).
- It can print the new data from the exception data record.

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 Customization Guide*. The DFH\$MOLS program reads the SMF data and formats and prints it. 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 119.

### **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 127.

## **Sample job stream for the DFH\$MOLS programs**

Figure 27 on page 126 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(1)))
/*
/**
/** Select CICS/ESA monitoring records
/**
//RCSEL   EXEC PGM=DFHJUP
//STEPLIB DD DSN=CICSTS13.CICS.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,FLDLN=2,COND=E
END
/**
//PRNT    EXEC PGM=DFH$MOLS
//STEPLIB DD DSN=CICSTS13.CICS.SDFHLOAD,DISP=SHR
//INPUT   DD DSN=&&TEMP2,DISP=(OLD,DELETE),UNIT
//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 27. 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 80.
- 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.

**Note:** The CICS/MVS 2.1.2 version of DFH\$MOLS will process SMF 110 monitoring data from any CICS/MVS version 2 release. The CICS Transaction Server for OS/390 Release 3 version of DFH\$MOLS will process SMF 110 monitoring data from any CICS/ESA version 3 or later release. However, DFH\$MOLS from an earlier version or release will not process the monitoring data from a later version or release. You should always use the DFH\$MOLS from the highest version or release.

**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

```
#           Note: It is important not to confuse the term "generic applid" with "generic
#           resource name".

#           Remember that "generic" and "specific" applids apply only to systems that
#           use XRF. CICS systems that do not use XRF have only one applid, which is
#           the value defined in the "APPLID=3D" parameter.
#           • For XRF, a CICS system's generic applid is defined on the APPLID
#           system initialization parameter and is the name by which CICS is known
#           in the network. (That is, it is the name quoted by remote CICS systems,
#           on the NETNAME option of CONNECTION definitions, to identify this
#           CICS.)
#           • A CICS system's specific applid is used to distinguish between the pair
#           of XRF systems. It is the name quoted on a VTAM APPL statement, to
#           identify this CICS to VTAM.
#           • A CICS system's generic resource name is defined on the GRNAME
#           system initialization parameter, and enables CICS to become a member of
#           a VTAM generic resource group.

#           Note, in particular, that you cannot use both VTAM generic resources and
#           XRF. If you use VTAM generic resources, you should specify only one name
#           on the APPLID system initialization parameter.
```

You control the processing that the DFH\$MOLS program performs by specifying the following control statements (for more details about these statements, see page 129):

\* [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 98 for 1998.

#### **IGNORE**

APPLID=xxxxxxxx[,yyyyyyyy,...]  
PRCSTYPE=xxxxxxxx[,yyyyyyyy,...]  
TERMID=xxxx[,yyyy,...]  
TASKNO=,nnnnnnn,[nnnnnnn,...]  
TRANID=xxxx[,yyyy,...]  
USERID=xxxxxxxx[,yyyyyyyy,...]

You can use any of these IGNORE options in conjunction with SELECT statements to form SELECT/IGNORE groups (see the BREAK control statement).

#### **OPTION GMT | LOCAL**

Use this control statement to control various report formatting options.

#### **PRINT**

DIC,EXC,PER,ALL

#### **SELECT**

APPLID=xxxxxxxx[,yyyyyyyy,...]  
PRCSTYPE=xxxxxxxx[,yyyyyyyy,...]  
TERMID=xxxx[,yyyy,...]  
TASKNO=nnnnnnn,[nnnnnnn,...]  
TRANID=xxxx[,yyyy,...]  
USERID=xxxxxxxx[,yyyyyyyy,...]

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.

#### **UNLOAD{DDNAME=xxxxxxxx[,LOCAL]}**

Use this statement to unload the input performance class monitoring data into the fixed length record format.

**Note:** This control statement only applies to CICS Transaction Server for OS/390 Release 2 and CICS Transaction Server for OS/390 Release 3 SMF records.

## 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**

specifies the end of 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, TRANID, PRCSTYPE and TASKNO 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**

specifies the number of records you want to process. The STOPAFT=nnnnnnnn parameter limits 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**

specifies the start and stop dates 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**

specifies 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**

specifies 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.

#### **Notes:**

1. CICS dictionary records are always processed by the DFH\$MOLS program and are not affected by any date/time period specification.
2. You do not have to specify both START and STOP; you can specify START without STOP, and STOP without START.
3. If you omit the DATE statement, records for all dates present in the input file are processed.
4. You can specify only one DATE statement (and associated TIME statement) in SYSIN.

#### **IGNORE [APPLID | PRCSTYPE | TASKNO | TERMID | TRANID | USERID]**

specifies that all records are to be excluded that have the specified generic

APPLID, CICS BTS process type, task number, or all records that have a specified transaction, terminal, or user identifier.

**APPLID=xxxxxxx[,yyyyyyyy,...**

Specify one or more generic APPLIDs to exclude monitoring data from a CICS region, or regions.

**PRCSTYPE=xxxxxxx[,yyyyyyyy,...**

Specify one or more 8-character BTS process-type identifiers, to exclude monitoring data associated with these process-types.

**TASKNO=nnnnnnn[,nnnnnnn,...**

Specify one or more task numbers to exclude monitoring data associated with these tasks.

**TERMID=xxx[,yyyy,...**

Specify one or more terminal identifiers to exclude monitoring data associated with these terminals.

**TRANID=xxx[,yyyy,...**

Specify one or more transaction identifiers to exclude monitoring data for these transactions.

**USERID=xxxxxxx[,yyyyyyyy,...**

Specify one or more user identifiers to exclude monitoring data for transactions submitted by these users.

You can specify each of the APPLID, PRCSTYPE, TASKNO, TERMID, 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 TERMID=, 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

**OPTION {GMT|LOCAL}**

specifies various DFH\$MOLS report formatting options.

**GMT** The DFH\$MOLS sample program is to print the monitoring record start and stop timestamp fields in GMT time in the reports produced.

**LOCAL**

The DFH\$MOLS sample program is to convert the monitoring record start and stop timestamp fields into local time in the reports produced.

**PRINT {ALL | DIC | EXC | PER | EXC,PER}**

specifies 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.

**DIC** lists only the monitoring performance class dictionary records that are selected by any other control statement options.

**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:** The SMF headers, SMF product sections, and CICS dictionary records are always printed except if the UNLOAD control statement is specified. In this case, the PRINT control statement must be specified to print the monitoring data required.

**SELECT [APPLID | PRCSTYPE | TERMID | TASKNO | TRANID | USERID]**

specifies the selection of all records of the specified generic APPLIDs, process—types, task numbers, transaction, terminal, or user identifiers.

**APPLID=xxxxxxx[,yyyyyyyy,...**

Specify one or more generic APPLIDs to include monitoring data from the CICS regions identified by these APPLIDs.

**PRCSTYPE=xxxxxxx[,yyyyyyyy,...**

Specify one or more CICS BTS process types to include monitoring data associated with these CICS BTS process types.

**TASKNO=nnnnnnn[,nnnnnnn,...**

Specify one or more task numbers to include monitoring data associated with these tasks.

**TERMID=xxx[,yyyy,...**

Specify one or more terminal identifiers to include monitoring data associated with these terminals.

**TRANID=xxx[,yyyy,...**

Specify one or more transaction identifiers to include monitoring data for these transactions.

**USERID=xxxxxxx[,yyyyyyyy,...**

Specify one or more user identifiers to include monitoring data for transactions submitted by these users.

You can specify each of the APPLID, PRCSTYPE, TASKNO, 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**

specifies 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 R14*.)

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 or when the UNLOAD control statement is specified.

#### **TIME**

specifies the start and stop times 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**

The start time of the period for which you want records processed, in the form hh.mm.ss or hhmss. A start time is optional, and if omitted the report includes all records for the start date, irrespective of time.

**STOP=stop-time**

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.

**Note:** CICS dictionary records are always processed by the DFH\$MOLS program and are not affected by any time period specification.

**TIMEOFF**

specifies the suppression of 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.

**UNLOAD {DDNAME=xxxxxxx[,LOCAL]}**

specifies that the performance class monitoring data is to be unloaded into a fixed length record format. The format of the output dataset can be mapped using the copy member DFHMNPDA provided in CICSTS13.CICS.SDFHSAMP.

**DDNAME**

The ddname for the output data set for the unloaded performance class records.

The default ddname of SYSUT4 is used if you do not code this keyword, and a SYSUT4 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.

**LOCAL**

The DFH\$MOLS sample program is to convert the performance class start and stop timestamp fields into local time in the output performance class records.

**Note:** If the UNLOAD control statement is specified, the SORT control statement must also be specified.

---

## 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            ERROR - INCOMPATIBLE CONTROL  
CARD OPTIONS REQUESTED**

**Explanation:** Some of the selected control statement options are in conflict. The DFH\$MOLS program has detected that the stop date specified on the DATE control statement is before the start date on the DATE control statement.

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 119 and "Monitoring dictionary records" on page 125.)

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.

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.

---

**112 There is no message text, but MVS abends U112 without a dump.**

**Explanation:** The DFH\$MOLS program was unable to open the data set specified on the SYSPRINT DD statement. Ensure that the JCL for the job is correct. A sample set of JCL to execute the DFH\$MOLS sample program is contained in Figure 27 on page 126.

---

**113 UNABLE TO OPEN DDNAME 'INPUT'; REPORT IS TERMINATED**

**Explanation:** The DFH\$MOLS program was unable to open the data set specified on the INPUT DD statement. Ensure that the JCL for the job was correct. A sample set of JCL to execute the DFH\$MOLS sample program is contained in Figure 27 on page 126.

This message is followed by an MVS abend U113 without a dump.

---

**114 UNABLE TO OPEN DDNAME 'xxxxxxx'; REPORT IS TERMINATED**

**Explanation:** The DFH\$MOLS program was unable to open the data set specified on the DD statement used for the UNLOAD control statement. 'xxxxxxx' is either SYSUT4, the default, or the ddname specified by the DDNAME= parameter on the UNLOAD control statement. Ensure that the JCL for the job was correct. A sample set of JCL to execute the DFH\$MOLS sample program is contained in Figure 27 on page 126.

This message is followed by an MVS abend U114 without a dump.

---

**115 ERROR - INCOMPATIBLE CONTROL CARD OPTIONS REQUESTED**

**Explanation:** The DFH\$MOLS program has detected that the UNLOAD control statement has been specified but the required SORT control statement has not been specified.

This message is followed by an MVS abend U115 without a dump.



---

## Chapter 13. 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
- LIST a specific APAR
- REMOVE a single group from a list on the CSD file
- SCAN all IBM-supplied groups and user defined groups for a resource. The definition of the matched resource in an IBM supplied group is compared to the definition(s) of the corresponding matched resource in the user groups.
- 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.

Note that the DFHCSDUP utility opens the CSD in non-RLS mode (even if you request RLS access on your JCL). This means that, if you access the CSD from CICS in RLS mode, it cannot be open when you run DFHCSDUP. The reason for the restriction is that the DFHCSDUP utility does not have the capabilities that are needed in order to open a recoverable file in RLS mode. The restriction also applies, however, if your CSD is nonrecoverable.

You can invoke the DFHCSDUP program in two ways:

1. As a batch program (see page “Invoking DFHCSDUP as a batch program” on page 141.)
2. From a user program running either in batch mode or in a TSO environment (see page 144).

---

## Sharing the CSD between CICS Transaction Server for OS/390 Release 3 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 Transaction Server for OS/390 Release 3.

In CICS Transaction Server for OS/390 Release 3, 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 Transaction Server for OS/390 Release 3, so you can safely update resource definitions from a CICS Transaction Server for OS/390 Release 3 region. If you are sharing the CSD between a CICS Transaction Server for OS/390 Release 3 region and a CICS/MVS 2.1.2 or a CICS/OS/VS 1.7 region, you can use the CICS Transaction Server for OS/390 Release 3 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 29 on page 142.)

The *CICS Transaction Server for OS/390 Migration Guide* discusses these obsolete attributes and their compatibility with earlier releases.

**Note:** You cannot use the EXTRACT command of the CICS Transaction Server for OS/390 Release 3 DFHCSDUP utility when the COMPAT option is specified.

---

## Input to the DFHCSDUP program

Input to the DFHCSDUP program (see Figure 28 on page 141) 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 28 on page 141) may be an updated primary file, or a print file.

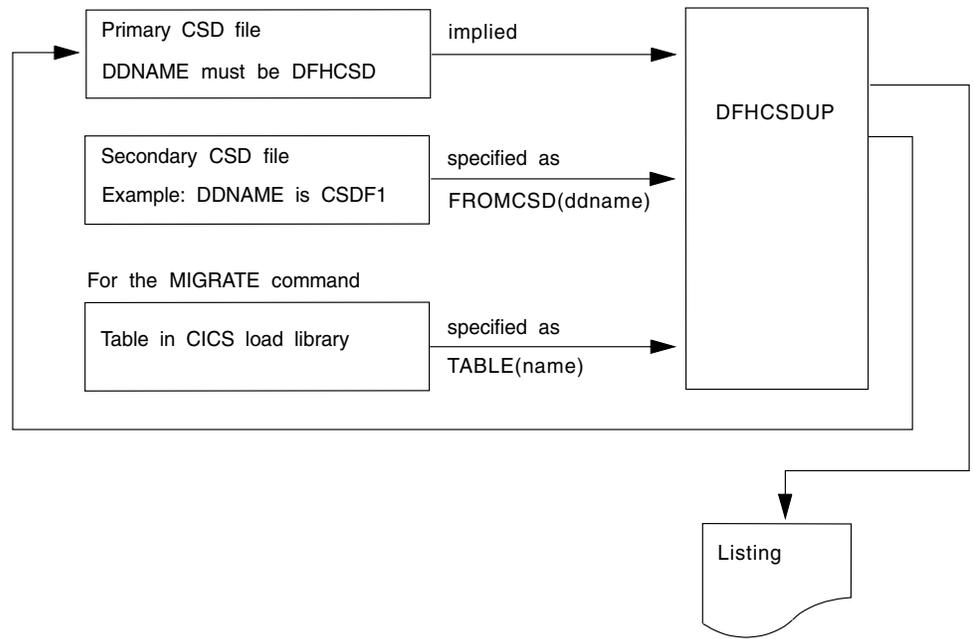


Figure 28. The DFHCSDUP offline utility program

---

## Invoking DFHCSDUP as a batch program

The job in Figure 29 on page 142 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=CICSTS13.CICS.SDFHLOAD,DISP=SHR
//*****
//* If you are running DFHCSDUP with the MIGRATE command,
//* and your CICS load tables are not in CICSTS13.CICS.SDFHLOAD,
//* concatenate your own private library here:
//*****
// DD DSN=CICSTS13.CICS.userlib.tables,DISP=SHR
//DFHCSD DD UNIT=SYSDA,DISP=SHR,DSN=CICSTS13.CICS.DFHCSD
//SECNDCSD DD UNIT=SYSDA,DISP=SHR,DSN=CICSTS13.CICS.SECNDCSD
//indd DD UNIT=SYSDA,DISP=SHR,DSN=extract.input.dataset
//outdd DD UNIT=SYSDA,DISP=SHR,DSN=extract.output.dataset
5
//* or
5
//outdd DD SYSOUT=A
5
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
:
:
DFHCSDUP commands
:
/*
//

```

Figure 29. 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 Transaction Server for OS/390 Release 3). The default is NOCOMPAT, which means that you cannot update obsolete attributes. For further information about this option, see “Sharing the CSD between CICS Transaction Server for OS/390 Release 3 and earlier releases” on page 140.

**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 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 9. 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	//CBDOUT DD SYSOUT=A
	SYSABOUT	//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 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 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 Transaction Server for OS/390 Release 3). The default is NOCOMPAT, which means that you cannot update obsolete attributes. For further information about this option, see “Sharing the CSD between CICS Transaction Server for OS/390 Release 3 and earlier releases” on page 140.

**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.

---

## Chapter 14. 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 30 on page 148. 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 30. 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 the *CICS 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

Add a group to a list.

### ADD syntax

▶▶—Add —Group(*groupname*)—List(*listname*)—▶▶

## Options

### Group(*groupname*)

specifies 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 do not specify a group, the current group name is added.

### List(*listname*)

specifies 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.

## Examples

To create a list LA01, by adding a group to it

```
ADD GROUP(GA001) LIST(LA01)
```

To add another group to list LA01

```
ADD GROUP(GA002) LIST(LA01)
```

LA01 now looks like this

```
GA001
```

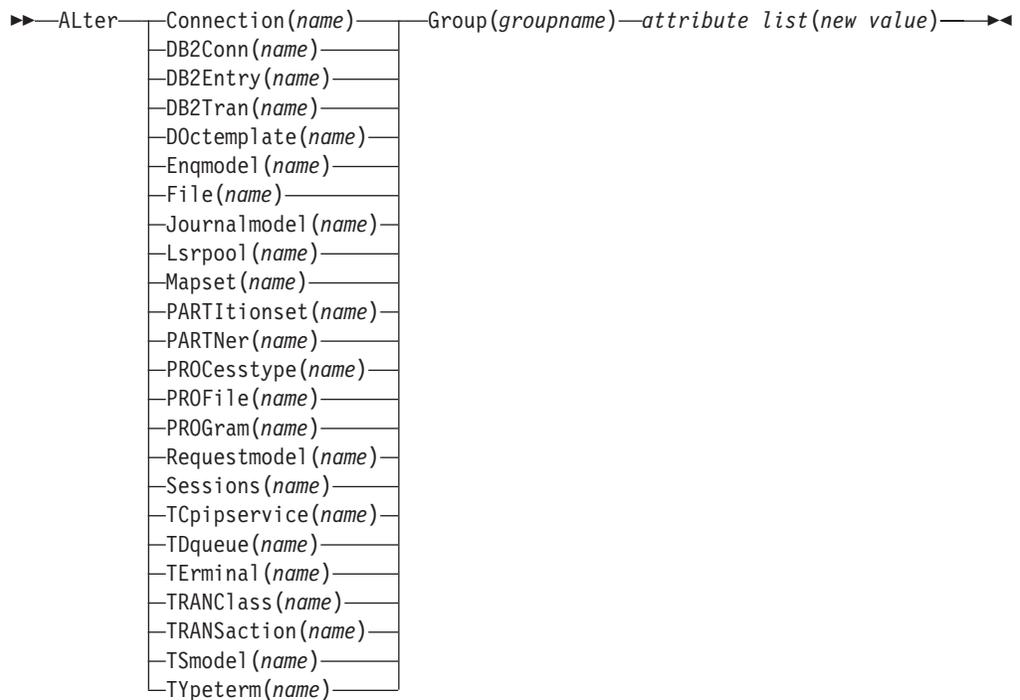
```
GA002
```

## ALTER

## ALTER

Change some or all of the attributes of an **existing** resource definition.

### ALTER syntax



## Description

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 Resource Definition Guide*.

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, delete the definition and define a new one with the correct values.

You can specify null attribute values, for example:

```
ALTER FILE(TEST) GROUP(ACT1) DESCRIPTION()
```

If an attribute for which you have specified a null value has a default, the value used depends upon the type of field:

- The command:

```
ALTER FILE(TEST) GROUP(ACT1) RLSACCESS() DESCRIPTION()
```

uses the default value of NO for RLSACCESS and the description is blanked out.

- The command:

```
ALTER FILE(TEST) GROUP(ACT1) PROFILE()
```

uses the default value DFHCICSA for the PROFILE field.

Changes to resource definitions in the CSD file do not take effect, in a running CICS system, until you install the group in which the resource definition resides.

## **REQTEXT**

**Generic naming in the ALTER command:** The ALTER command accepts both generic resource names and group names.

For each resource in the CSD file matching the specified combination of resource name and group name, an ALTER is attempted. In the case of an individual ALTER failing, processing terminates when all attempts for the command have been processed.

## **Options**

### **Attribute list**

specifies the attributes to be altered.

### **Group**(*groupname*)

specifies the name of the group containing the resource to be altered.

### **Resource**(*name*)

specifies the resource whose attributes you want to alter. You can specify a generic name by using the characters + and \*.

## **Examples**

To make a program resident:

```
ALTER PROGRAM(ERR01) GROUP(GENMODS) RESIDENT(YES)
      DATALOCATION()
```

To make all programs in the group GENMOD resident:

```
ALTER PROGRAM(*) GROUP(GENMOD) RESIDENT(YES)
      DATALOC()
```

---

**APPEND**

Add the groups in one list to the end of another list.

**APPEND syntax**

►►—APpend—FRomcsd(*ddname*)—LIst(*listname1*)—To(*listname2*)—►►

**Description**

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.

**Options****FRomcsd**(*ddname*)

specifies the *ddname* of the secondary CSD file from which you are appending *listname1*.

**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 file, or on another CSD file. If you are appending from another CSD file, you must identify it by specifying the FROMCSD parameter.

**To**(*listname2*)

specifies the name of the list to which you want the group names appended. If you are appending from another CSD file, 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.)

**Examples**

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



**MErge**

If *groupname2* already exists and duplicate definitions occur, the original definitions in *groupname2* are preserved.

**Replac**

If *groupname2* already exists and duplicate definitions occur, the definitions in *groupname1* replace those in *groupname2*.

**To(*groupname2*)**

specifies the name of the group to which the definitions are copied. If you are copying from another CSD file, 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 154 for details.

**Examples**

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:

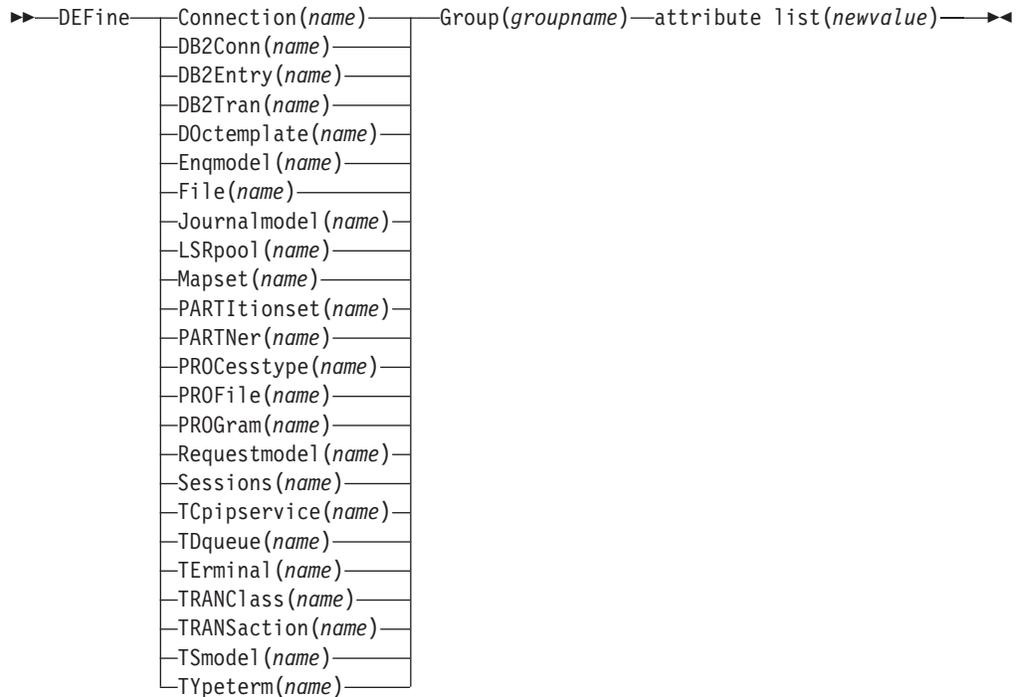
```
COPY GROUP(ABCD*) TO(NEWGROUP)
```

## DEFINE

## DEFINE

Create new resource definitions.

### DEFINE syntax



## Options

### 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, *CICS Resource Definition Guide*. Attributes that you do not specify are given default values.

### Group(groupname)

specifies 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.

### Resource(name)

specifies the name of the resource you want to define. Do not use a generic resource name. The resource option must always be the first operand of the DEFINE command.

## Examples

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)
```

## DEFINE

The next 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)
```

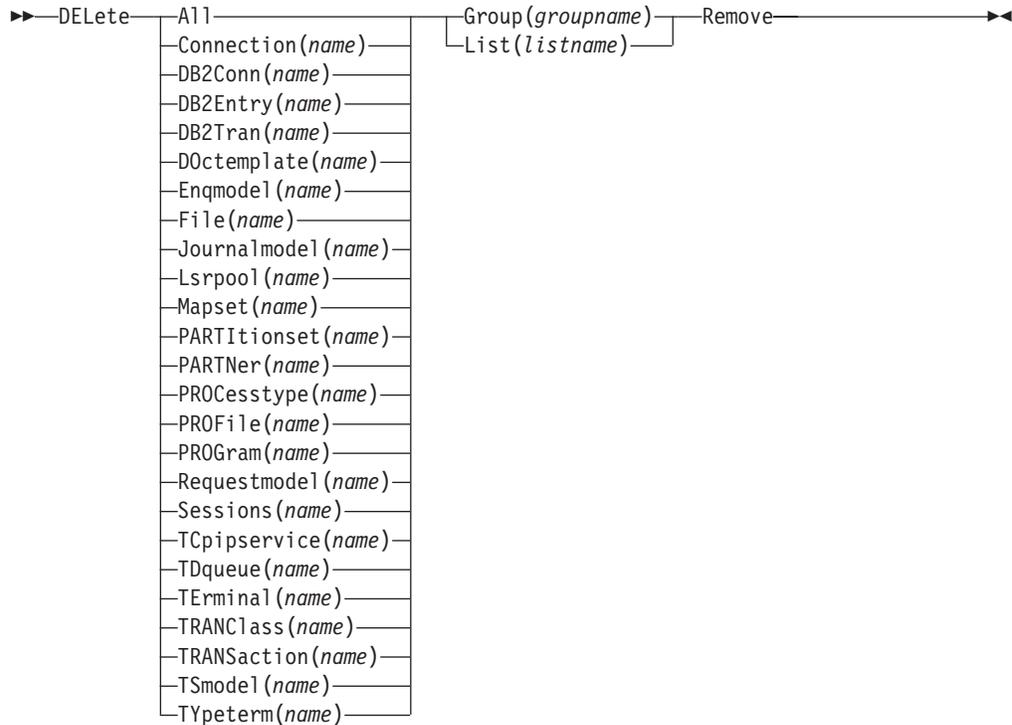
The INITIALIZE command generates a TYPETERM definition, but not a TERMINAL definition, for a console. You must have at least one console defined in order to issue MVS MODIFY commands to CICS. Console id 00 is used to issue commands using MVS job control language, and by authorized programs that use that MGCR macro to issue MVS commands.

## DELETE

## DELETE

Delete a single resource definition in a group, all the resource definitions in a group, or all the group names in a group list.

### DELETE syntax



## Description

Deleting a resource definition is different from removing a group from a list (see "REMOVE" on page 170). A deleted resource definition really does disappear from the CSD file.

### Note:

When you DELETE the last resource in a group, the group is automatically deleted. An empty group cannot exist.

When a group is deleted, the group is removed from all lists that had contained it, except when running UPGRADE commands.

You cannot 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.

## Options

### 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.

**List**(*listname*)

specifies the name of the list to be deleted. Do not use a generic list name.

**Remove**

If this is specified when the group is deleted, the group is removed from all lists that contained it unless UPGRADE commands are running.

**Resource**(*name*)

specifies the name of the resource to be deleted. Do not use a generic resource name.

This operand can be used only with the GROUP option.

## Examples

A list in the primary CSD file 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:

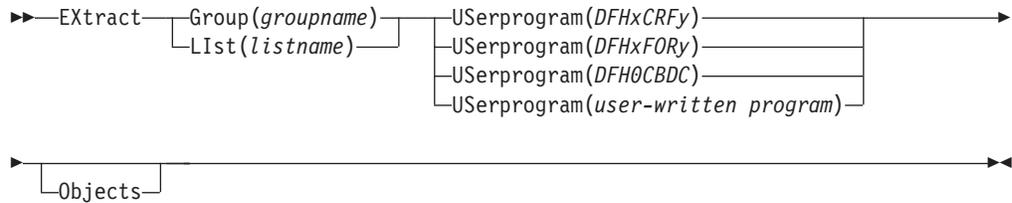
```
DELETE LIST(LISTA)
```

**Note:** The resource definitions in the groups in LISTA are not deleted.

## EXTRACT

Extract a resource definition, group, or list from the CSD file.

### EXTRACT syntax



## Description

You can use the EXTRACT command to extract resource definition data from the CSD file, 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 Customization Guide*.

## Options

### Group(*groupname*)

specifies only those resource definitions within the named group. You can specify a generic group name.

### LIst(*listname*)

specifies 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*)

specifies 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

## EXTRACT

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 CICSTS13.CICS.SDFHSAMP, and the assembler versions are also available in pregenerated form in CICSTS13.CICS.SDFHLOAD.

### Examples

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
```

## INITIALIZE

---

## INITIALIZE

Prepare a newly defined data set for use as a CSD file.

### INITIALIZE syntax

►►—INITialize —◄◄

### Description

You must initialize your CSD file before you can use any of the other DFHCSDUP commands, or the RDO transactions. After you have initialized your CSD file, you do not need to execute this function again.

The standard entries for the CICS-supplied resource definitions are created on the CSD file. 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.

CICS supports RDO for transient data. The DFHDCTG group contains sample definitions of all the CICS-supplied queues. You can add the names of other queues that you want to be installed at the same time to DFHDCTG. Place DFHDCTG at the top of DFHLIST so that the queues become available for use at the earliest possible point during CICS initialization.

If you use another group to install the CICS-supplied queues, make sure that this group is at the top of the first list to be installed using GRPLIST as part of an initial or cold start.

You can put other transient data resource definitions into different groups, from which they can be installed either during an initial or cold start, or at some point after initialization has completed.

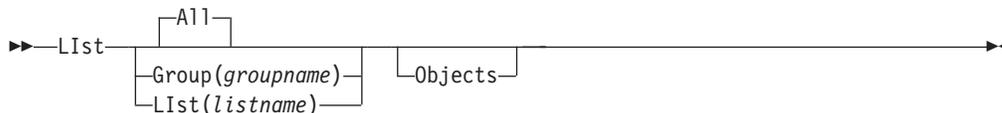
INITIALIZE also creates a control record at the start of the CSD file. This record contains fields identifying the CICS release and the current level of service applied to the CSD. It also has fields containing the date and time of creation of the CSD file, and the date and time the file was last updated. Both these fields appear on the hard copy listing of the CSD file produced by the LIST command.

If you want to prepare a newly defined recoverable data set for use as a CSD file, you must INITIALIZE it using non-RLS mode, because a recoverable data set cannot be opened for output from batch in RLS mode, but the data set needs to be opened for output in order to initialize it.

## LIST

Produce listings of the current status of the CSD file.

### LIST syntax



## Description

The listings are output to the SYSOUT 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.

## Options

### Group(*groupname*)

specifies only those resource definitions within the named group. You can specify a generic group name.

### List(*listname*)

specifies 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

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.

## 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 file.

- 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 file, together with the properties of the resources in all the groups.

## LIST

- 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

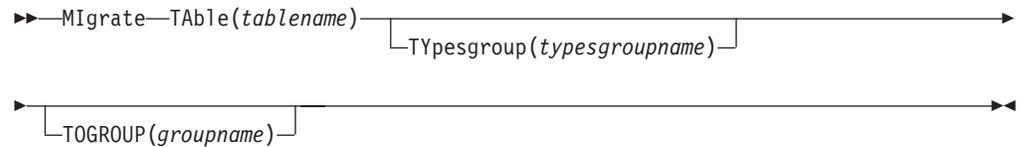
This enables you to tabulate the properties of all the resources to be defined in a CICS system at startup time. These are identified by the list name or names specified in the GRPLIST=(*list1,list2,list3,list4*) system initialization parameter. The names of all the groups in the list appear in the summary 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

Transfer the contents of a DCT, an FCT, an RCT, a TCT, or a TST, from a CICS load library to the CSD file.

### MIGRATE syntax



## Description

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.

- **To transfer a DCT**, the format is:

```
MIGRATE TABLE(tablename) TOGROUP(groupname)
```

where TABLE(*tablename*) identifies the name of the table in the load library (DFHDCTxx).

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 loaded. For migration purposes, DCTs must be link-edited with AMODE(24) RMODE(24). To ensure this, you must specify a DFHDCT TYPE=(INITAL,MIGRATE) statement in your DCT—failure to do so causes the DFHDCT macro to force AMODE(31), which results in errors when running DFHCSDUP.

The result is a set of groups containing TDQUEUE resource definitions. You can specify each group using the macro:

```
DFHDCT TYPE=GROUP, GROUP=xxxxxxxx
```

which you insert in the DCT source instructions before you assemble them for migration. All definitions **after** such a TYPE=GROUP macro (up to the next TYPE=GROUP macro) go into the group named by GROUP=xxxxxxxx. Definitions that occur **before** the first such TYPE=GROUP macro are migrated to the default group. You can also specify that definitions are to be migrated to the default group by inserting the following macro in the DCT before the definition entries:

```
DFHDCT TYPE=GROUP, GROUP=*DEFAULT
```

You can use the TOGROUP parameter of the MIGRATE command to assign a specific name to the default group. If you do not specify TOGROUP, the name of the default group is taken from the table name. For example, if the migrated table name is DFHDCT24, the name of the group created is DCT24.

- **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 using the macro:

## MIGRATE

```
DFHFCT TYPE=GROUP, GROUP=xxxxxxx
```

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 an RCT**, the format is:

```
Migrate TABLE(tablename) [TOGROUP(groupname)]
```

where TABLE(*tablename*) identifies the name of the table in the load library (DFHRCTxx).

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 loaded. For migration purposes, RCTs must be link-edited with RMODE(24).

The result is a set of groups containing DB2CONN, DB2ENTRY and DB2TRAN resource definitions. You can define each group using the macro:

```
DSNCRCT TYPE=GROUP, GROUP=xxxxxxx
```

which you insert in the RCT source instructions before you assemble the RCT for migration. All definitions **after** such a TYPE=GROUP macro (up to the next TYPE=GROUP macro) go into the group named by GROUP=xxxxxxx. Definitions that occur **before** the first such TYPE=GROUP macro are migrated to the default group. You can also specify that definitions are to be migrated to the default group by inserting the following macro in the RCT before the definition entries:

```
DSNCRCT TYPE=GROUP, GGROUP=*DEFAULT
```

You can use the TOGROUP parameter of the MIGRATE command to assign a specific name to the default group. If you do not specify TOGROUP, the name of the default group is taken from the table name. For example, if the table name is DFHRCT24, the name of the group created is RCT24.

- **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 using the macro:

```
DFHTCT TYPE=GROUP, GROUP=xxxxxxx
```

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 file.

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 file. (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 file 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.

**To transfer a TST**, the format is:

```
Migrate TABLE(tablename) [TOGROUP(groupname)]
```

where TABLE(*tablename*) identifies the name of the table in the load library (DFHTSTxx) and TOGROUP(*groupname*) specifies the name of the group to contain the definitions obtained from the TST.

The content of a table is transferred as a group containing TSMODEL definitions. When migrating large tables, make sure that you allocate a sufficiently large region for the largest table.

For migration purposes, TSTs must be link-edited with AMODE(24) RMODE(24). To ensure this, you must specify a DFHTST TYPE=(INITIAL,MIGRATE) statement in your TST. Failure to do so causes the DFHTST macro to force AMODE(31), which leads to errors when running DFHCSDUP.

You can use the TOGROUP parameter of the MIGRATE command to assign a specific name to the default group. If you do not specify TOGROUP, the name of the default group is taken from the TABLENAME. For example, if the tablename is DFHTSTJP, the name of the group created is TSTJP.

**Note:** TSMODEL definitions have a location attribute, either Main or Auxiliary. Migration will always set this to auxiliary (although you can change it later by updating the TSMODEL definition).

## MIGRATE

Before you define TSMODEL resource definitions to replace TST macros, you are able to specify *Main* or *Auxiliary* on the WRITEQ TS API command, but this is ignored if a TSMODEL resource definition with a matching prefix is installed in the system; the value supplied by the TSMODEL is used instead.

### Options

**Table**(*tablename*)

specifies the name in the load library of the table you want to migrate (that is, DFHDCTxx, DFHFCTxx, or DFHTCTxx).

**TOgroup**(*groupname*)

specifies the name of the group to which the definitions are to be migrated. This is for use with DCT migration only.

**TYPESgroup**(*typesgroupname*)

specifies the name of the group to which the TYPETERM definitions are to be migrated. For use with TCT migration only.

---

## PROCESS

Apply maintenance to the CSD file for a specific APAR.

### PROCESS syntax

▶▶—PROCESS—Apar(*aparnumber*)————▶▶

## Description

The PROCESS APAR command is used to apply maintenance to your CSD file for a specific APAR. Only use this command in accordance with the instructions in the associated PTF cover letter.

## Options

### Apar(*aparnumber*)

The number of the APAR providing the maintenance; for example, PROCESS APAR(PQ12417) is used to apply maintenance for APAR PQ12417.

## REMOVE

---

## REMOVE

Remove a group name from a list.

### REMOVE syntax

▶▶—Remove —Group(*groupname*)—List(*listname*)————▶▶

## Description

The group, and all its resource definitions, still exists on the CSD file.

## Options

### Group(*groupname*)

specifies the name of the group to be removed. Do not use a generic group name.

### List(*listname*)

specifies 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 file.

## Examples

A list LL02 contains the following groups:

G001 G002 G003 G004

To remove group G003:

```
REMOVE GROUP(G003) LIST(LL02)
```

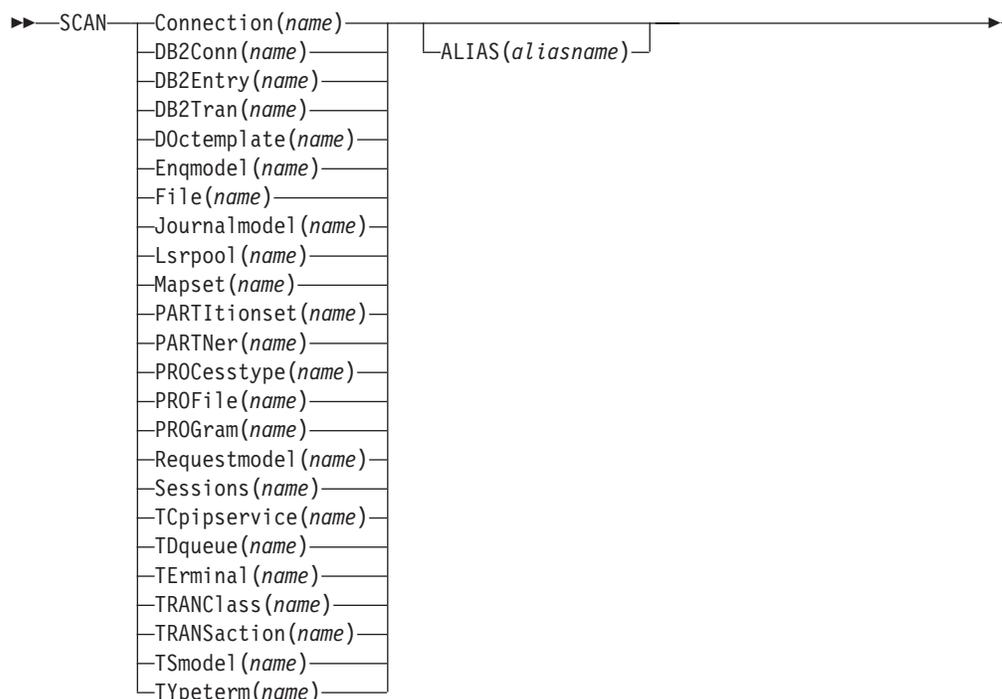
This leaves:

G001 G0023 G004

## SCAN

SCAN all the IBM-supplied groups and user-defined groups for a specified resource. The definition of the matched resource in an IBM supplied group is compared with the definition(s) of the corresponding matched resource in the user groups.

### SCAN syntax



## Description

For information about the types of resource that you can specify on the SCAN command, and for a description of the attributes and default values of each resource type, see the *CICS Resource Definition Guide*.

The SCAN command searches all the IBM supplied groups in the CSD for a resource definition of a specified name and type. A message is issued with the results of the search. The user-defined groups are then searched for the same resource definition. The outcome of this can be one of the following:

- If an IBM-supplied group and one or more user-defined groups contain the resource definition, a comparison is made between the definition in the IBM-supplied group and the user group(s). A message is issued indicating whether the definition in the IBM supplied group matches the definition(s) in the user group(s).
- If the resource definition is not found in the user defined groups a message is issued.
- If the resource definition is not found in an IBM-supplied group but is found in one or more user defined groups a message is issued indicating the group(s) that contained it.

If *aliasname* is specified, the user groups are searched using *aliasname*.

## SCAN

### Notes:

1. The compatibility groups DFHCOMPx are not scanned as part of the IBM supplied groups but as user defined groups.
2. The DESCRIPTION attribute is not used in the comparison.

You can use the SCAN command to check for differences between IBM-supplied definitions that you have modified and the latest IBM-supplied versions after an upgrade.

## Options

### **Alias**(*aliasname*)

specifies the alias name of the resource type to be searched for in the user-defined groups.

This operand is optional.

### **Resource**(*name*)

specifies the name of the resource type to be searched for in the IBM-supplied groups, and in the user-defined groups if *aliasname* is not specified. The resource option must always be the first operand of the SCAN command.

## Examples

To search the CSD for transaction CEDA:

```
SCAN TRANSACTION(CEDA)
```

The result of this could look like:

```
DFH5130 I PRIMARY CSD OPENED; DDNAME: DFHCSD
DFH5633 I TRANSACTION CEDA FOUND IN GROUP DFHSPI
DFH5631 I TRANSACTION CEDA FOUND IN GROUP A1
          MATCHES THE IBM SUPPLIED DEFINITION IN GROUP DFHSPI
DFH5631 I TRANSACTION CEDA FOUND IN GROUP A2
          MATCHES THE IBM SUPPLIED DEFINITION IN GROUP DFHSPI
DFH5632 I TRANSACTION CEDA FOUND IN GROUP DFHCOMP1
          DOES NOT MATCH THE IBM SUPPLIED DEFINITION
          IN GROUP DFHSPI
DFH5101 I SCAN COMMAND EXECUTED SUCCESSFULLY.
```

To search the CSD for transaction CEDA with an alias name of AEDA:

```
SCAN TRANSACTION(CEDA) ALIAS(AEDA)
```

The result of this could look like:

```
DFH5130 I PRIMARY CSD OPENED; DDNAME: DFHCSD
DFH5633 I TRANSACTION CEDA FOUND IN GROUP DFHSPI
DFH5631 I TRANSACTION AEDA FOUND IN GROUP A3
          MATCHES THE IBM SUPPLIED DEFINITION IN GROUP DFHSPI
DFH5101 I SCAN COMMAND EXECUTED SUCCESSFULLY.
```

---

## SERVICE

Carry out maintenance to your CSD file.

### SERVICE syntax

►►—Service—FRomcsd(*ddname*)—LEve1(*nnn*)—◄◄

## Description

You might occasionally (between CICS releases) have to apply a service routine to carry out preventive or corrective maintenance to your CSD file. 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 file, from the existing CSD file. All the definitions are preserved, with the corrections (if any) applied.

## Options

### FRomcsd(*ddname*)

specifies the *ddname* of the current CSD file, which for the purposes of the command is treated as the secondary CSD file.

### LEve1(*nnn*)

Associated with your CSD file 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 file 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

## UPGRADE

Change the CICS-supplied resource definitions in a primary CSD file.

### UPGRADE syntax

```
►►—UPgrade—┬──USing(filename)──┬──Replace──┬──►►
```

## Description

Upgrades the IBM-supplied definitions in the CSD. Definitions are added to, modified in, or deleted from DFH-groups. Note that deleted definitions are added to compatibility groups with names of the form DFHCOMP*n*. This enables you to share the CSD with earlier releases of CICS after you have run the upgrade command.

The upgrade command can also be used to apply any package of IBM-supplied resource definitions to the CSD file. For example, the definitions for the CICS sample programs and transactions can be transferred to the CSD file with the UPGRADE statement.

## Options

### Replace

Specify the REPLACE option when you need to rerun the UPGRADE command (for example, because of a previous failure).

### USing(*filename*)

Upgrading a CSD file does not require you to use the USING operand. All IBM-supplied definitions from **any** release are deleted and then the CSD file 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 file.

---

## VERIFY

Remove internal locks on groups and lists.

### VERIFY syntax

►►—VERIFY —————►►

## Description

Use the VERIFY command only when the CSD file is not in use and no backout processing is pending on the CSD file; preferably use it only when no CICS systems that may use the CSD file are running. In particular, do not use the VERIFY command while CICS systems could be accessing the CSD file in RLS access mode.

VERIFY acts on the whole CSD file, 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 file 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 file.

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.

## VERIFY

---

## Chapter 15. Load module scanner (DFHEISUP)

# This chapter describes the load module scanner. This utility can scan load libraries  
# for the CICS commands in load modules, and identify which modules contain  
# certain API commands.

---

### # Overview

# The load module scanner acts as a batch utility by scanning load modules for CICS  
# API commands. It scans modules and reports back if they contain the specified  
# commands. The load module scanner locates all commands and provides a filter  
# which allows you to report on only the commands in which you are interested.

# The commands to be reported upon are specified as a filter input file. The modules  
# to scan can be specified as an entire PDS, or a list of modules. A sample command  
# list will be provided to search for programs which are not supported to run in the  
# 3270 bridge environment. The output report will specify which modules contained  
# which commands, at which offsets, including EDF information if available.

# The load module scanner provides the return of a report containing any located,  
# requested commands.

# You can use the load module scanner to produce a summary report or a detailed  
# report of modules that contain the EXEC CICS commands you are interested in.  
# Figure 31 shows how the scanner works.

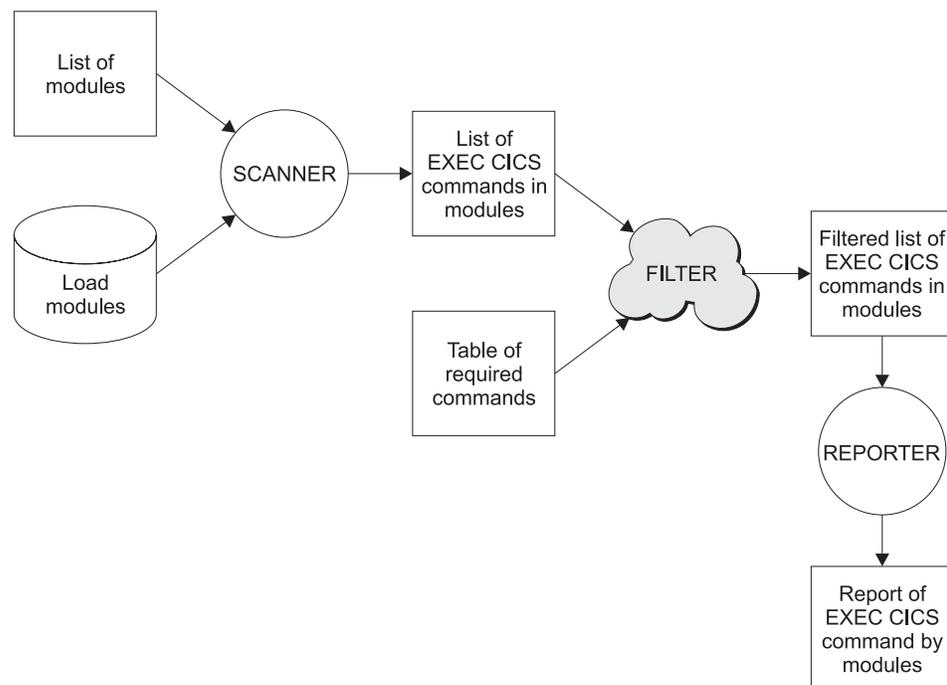


Figure 31. Load module scanner with filter

# Two options, summary and detailed, depending on the PARM command, can be  
# specified. These are separately documented.

---

## # Creating a summary report

# You can request a summary report from the load module scanner by editing and  
# running the job DFHEILMS JCL. Figure 32 on page 179 shows an example of a  
# summary scan which includes the following additional statements:

### # DFHFLTR

# The input file for the filter, specifying which commands are being sought.

### # DFHDTL

# An optional dataset to receive a list of modules in which the specified  
# commands were located. This can then be used in the DFHLIST step of a  
# detailed scan.

# You can request a summary report from the scanner, editing the following as  
# appropriate:

- # • The JOB accounting parameters
- # • The PARM statement

#  
# 

#  
#

### # SUMMARY

# Specifies that a summary scan (and report) is required for the entire library,  
# except for CICS modules, CICS tables, and those modules that cannot be  
# loaded (due to an error).

### # DETAILMODS

# Specifies that the names of those modules containing at least one possible  
# requested EXEC CICS command are to be written to the sequential file defined  
# by the DFHDTL DD statement. This file may be used to restrict a subsequent  
# detailed report, by specifying it on the DFHLIST DD statement of a detailed  
# report run of the scanner. For more information about the DFHLIST DD  
# statement, see “Creating a detailed report” on page 179.

### # The STEPLIB DD statement

# Specifies the name of the load library where you have installed the scanner  
# program, DFHEISUP.

### # The DFHIN DD statement

# Specifies the name of the load library to be scanned. This must not contain  
# concatenated datasets. The DFHIN DD statement accepts only PDSs. If a PDSE  
# is supplied, the results are undefined.

### # The SYSPRINT DD statement

# Specifies the destination for the summary report.

### # The DFHDTL DD statement

# Specifies the name of the sequential data set where the list of modules with  
# potential command matches is to be sent. You can edit the data set to alter the  
# list of modules to be scanned before running the scanner to produce a detailed  
# report.

# Figure 32 on page 179 shows an example of a summary scan.  
#



# **DFHLIST**  
# An optional data set containing a list of modules in which previously specified  
# commands were located. For more information about the contents of this list,  
# see the description of the DFHLIST DD statement.

# The following statements should be edited as appropriate:

# **PARM statement:**

#  
#  
# 

#  
#  
# **DETAIL**  
# Specifies that a detailed scan is required. The extent of the scan is defined by  
# the ALL parameter.

# **ALL**  
# Specifies that all modules in the load library are to be scanned for requested  
# EXEC CICS commands.  
#  
# If ALL is omitted, only those modules listed in the file specified on the  
# DFHLIST DD statement are to be scanned. This file would normally be from  
# the DFHDTL DD output of a summary report run which you can edit before  
# creating a detailed report.

# **STEPLIB DD statement**  
# Specifies the name of the load library in which you have installed the scanner  
# program, DFHEISUP.

# **DFHIN DD statement**  
# Specifies the name of the load library to be scanned. This should not contain  
# concatenated datasets. See the note in the following section for more  
# information. The DFHIN DD statement accepts only PDSs. If a PDSE is  
# supplied, the results are undefined.

# **SYSPRINT DD statement**  
# Specifies the destination for the detailed report.

# **DFHLIST DD statement**  
# Specifies the name of the data set containing the list of modules to be scanned.  
# The list may be created initially as the output from a summary run of the  
# scanner. If you specify ALL on the PARM statement, change the DFHDTL DD  
# statement to specify //DFHLIST DD DUMMY.  
#  
# This file consists of the member names of each module within the PDS  
# specified by the DFHIN DD that are to be scanned. Each module name should  
# appear on a separate line.

# **Note:** During a detailed scan, the modules to be scanned are read from the DD  
# DFHLIST. This list was created as DD DFHDTL during a SUMMARY  
# DETAILMODS scan. When a DETAIL scan is being performed ensure that  
# DFHIN is set to the same location as when the SUMMARY DETAILMODS  
# was run. The scanner looks for the modules listed in DD DFHLIST in the  
# location specified by DD DFHIN.

#

```
# //DFHSCNR JOB (accounting information)
# //DFHSCAN EXEC PGM=DFHEISUP,PARM=('DETAIL'),REGION=512M
# //STEPLIB DD DSN=HLQ.SDFHLOAD,DISP=SHR
# //SYSPRINT DD SYSOUT=*
# //SYSERR DD SYSOUT=*
# /* Filter file, supplied by user, or from samples dataset
# //DFHFLTR DD DSN=HLQ.FILTER,DISP=SHR
# /* Module list, supplied by user, used during detail runs.
# //DFHLIST DD DSN=HLQ.MODLIST,DISP=SHR
# /* PDS containing loadmodules to be scanned
# //DFHIN DD DSN=HLQ.SDFHLOAD,DISP=SHR
```

```
#
# Figure 34. Example of a detailed scan
```

```
#
# Figure 35 on page 182 shows the contents of a detailed report.
```

```
#
```



# • Library totals, as with the summary report, but only for those modules selected  
 # for the detailed run.

# **Note:** For both detailed and summary reports the identified language of the  
 # module is based up the language of the first identified command, as with  
 # the CICS Affinities Load Module Scanner, DFHEISUP. If the module is a  
 # mixed language module, only the first located language will be reported. In  
 # addition, if the first identified command is a false hit, the language for that  
 # module will be reported incorrectly.

---

## # Filter input format definition

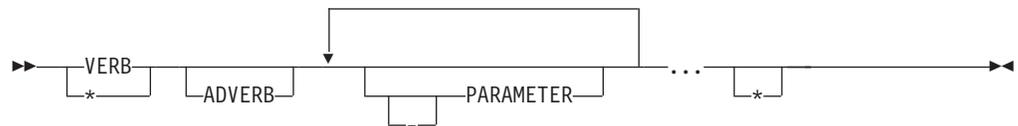
# In most cases, rather than just extract all EXEC CICS commands in load modules,  
 # you can look for a subset of commands — in particular, look for which modules  
 # contain these commands. For example, you might want to look for the commands  
 # which are not supported for certain usage in CICS. Each command that you want  
 # to find is placed on a separate line, for example:

- # • WRITEQ TS
- # • DELETEQ TS

# Then, additionally, parameters may be specified. When specifying parameters, two  
 # operators are allowed, denoting as follows:

# – not interested in commands containing this parameter

# \* any parameter



#

# Table 10 shows you how commands can be specified for filter input format  
 # definitions.

# *Table 10. Example of commands for filter input format definition*

Command	Parameter specification
WRITEQ TS –NUMITEMS SYSID *	All WRITEQ TS commands, without NUMITEMS, but with SYSID
WRITEQ TS – NUMITEMS *	All WRITEQ TS commands, without NUMITEMS
WRITEQ TS *	All WRITEQ TS commands, any parameters
WRITEQ TS QUEUE FROM LENGTH	All WRITEQ TS commands only with QUEUE & FROM & LENGTH
* LENGTH *	All commands with a LENGTH parameter
* *	All commands with any parameters

# The \* LENGTH \* line shows how the 'any' operator specification can be used to  
 # search for any command. The second \* operator is needed, otherwise the line  
 # would imply all commands with only a LENGTH parameter.

# In addition, comments may be placed in the file by beginning the line with a hash  
# character, #.

# Each filter specified in the file will be checked to ensure it represents a valid filter.  
# Checks are made to ensure the command exists, that the parameters exist, and that  
# they are usable as filter data. Some parameters cannot be identified by the scanner  
# — if they are specified as part of a filter, a warning message is returned. If a  
# command or parameter could not be found, you receive an error message, and the  
# load module scanner terminates with a return code of 12. If 'any' is specified for a  
# command, for example, \* LENGTH \*, no validation is performed upon the  
# parameters.

# A sample table 'DFHEIDBR' is supplied within the samples library. This library  
# provides the filter set for commands which are not supported by the 3270 bridge.

# Sample JCL is also supplied, DFHEILMS JCL, for executing the scanner with the  
# supplied sample filter input set.

# **Note:** The following exceptions apply to the load module scanner:

# **Misdetection of verbs**  
# SPOOLOPEN INPUT will be detected as SPOOLOPEN OUTPUT, and  
# WAIT JOURNAL will be detected as WAIT JOURNALNUM.  
# SPOOLOPEN OUTPUT and WAIT JOURNALNUM will, where present,  
# be detected as themselves.

# INQUIRE UOWENQ will be detected as INQUIRE ENQ.

# **Equivalent or CVDA—based parameters**  
# Where one parameter is really an equivalent of another, for example,  
# NOSUSPEND on ALLOCATE is equivalent to NOQUEUE, the preferred  
# keyword is returned.

# Where parameters can be coded as CVDA values, only the real  
# parameter will be located by the scanner. Additionally, if equivalent or  
# CVDA-based parameters are included within a filter file, the scanner  
# sends a warning to inform you which parameters cannot be located.

# Jusfirst and Juslast on SEND TEXT will be detected as JUSTIFY, and  
# reported as CVDA values.

# **False hits**  
# As with all types of scanners, it is not possible to guarantee that a hit is  
# genuine. There may be a few false hits due to static data in a module.  
# Therefore, filters can only be used to identify candidates matching a  
# given criteria, rather than providing a match.

---

## # **Activating trace**

# Trace should be enabled only at the request of your IBM service representative.

# You should prefix the PARMs in the JCL with the required trace type. One at a  
# time of the supported trace types, TI, TD and TB, can be specified.

```

//DFHSCNR JOB (accounting information)
//DFHSCAN EXEC PGM=SCANNER,PARM=('TB SUMMARY, DETAILMODS'),REGION=128M
//STEPLIB DD DSN=HLQ.LOADLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSERR DD SYSOUT=*
/* Filter file, supplied by user, or from samples dataset
//DFHFLTR DD DSN=HLQ.FILTER,DISP=SHR
/* Module list, to be created during summary,detail runs.
//DFHDTL DD DSN=HLQ.MODLIST,DISP=(NEW,CATLG,DELETE),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=8000),SPACE=(CYL,(1,1))
/* PDS containing loadmodules to be scanned
//DFHIN DD DSN=HLQ.DFHRPL,DISP=SHR

```

*Figure 36. Example of a summary scan modified to activate trace.*

#

Figure 36 shows an example summary run including the trace type TB.



---

## Chapter 16. Batch-enabling sample programs for RLS access-mode data sets

If you have recoverable VSAM data sets that are open in RLS access mode to CICS regions, and you need to update them from a batch application, you must first quiesce the data sets before using them in batch mode. This chapter describes some procedures that you can use to help you to automate the process of preparation for batch processing.

Before going on to discuss the process, we first discuss the reasons why such a process may be necessary.

---

### Switching from RLS to non-RLS access mode

A batch program cannot open a data set in non-RLS access mode if there are any files open against it in RLS access mode. To switch from RLS to non-RLS access mode for batch update, you must first quiesce the data set. The VSAM RLS quiesce mechanism causes all CICS regions in the sysplex to close any RLS-mode files that are open against a specified data set. After they have been closed under the quiesce mechanism, data sets can be opened only in non-RLS mode. To re-enable quiesced data sets to be re-opened in RLS mode, all open non-RLS mode files must be closed and then the data sets must be unquiesced.

**Note:** The quiesce mechanism cannot inform batch programs that have the data set open in RLS access mode about the quiesce request. If you have such programs, you should use the DFSMS™ SHCDS LIST subcommands to check whether any non-CICS jobs have files open in RLS mode against the data set. For information about the SHCDS LIST subcommand, see *DFSMS/MVS Access Method Services for ICF, SC26-4906*.

Quiescing a data set sets the quiesce flag in the ICF catalog so that the data set can be opened in non-RLS mode only. This is the recommended way of making data sets available for batch programs. However, even if a data set has been quiesced, you still cannot open it for update in non-RLS access mode if SMSVSAM is holding retained locks against the data set. This is because the locks are needed to preserve data integrity: they protect changes that are waiting to be either committed or backed out.

For more information about the procedures you should follow for checking and handling retained locks when switching to non-RLS mode, see the *CICS Recovery and Restart Guide*.

---

### Preparing data sets for batch operations

CICS provides a suite of eight sample application programs that are designed to help you to automate your batch preparation procedures for data sets that are opened in RLS mode.

You can use these sample programs unmodified, or you can use them as a basis for writing your own programs. The programs are DFH0BAT1 through DFH0BAT8.

Before attempting to run your batch jobs, you should ensure that:

- No retained locks are held for the data sets
- No files are open against the data sets in RLS mode.

The sample programs, using the INQUIRE DSNAME, INQUIRE UOWDSNFAIL, and SET DSNAME SPI commands, help you to deal with any retained locks. When you have successfully dealt with these, you can quiesce the data sets to close the RLS-mode files using the SPI or CEMT commands.

Three of the programs are coordinating programs, which use CICS distributed program link (DPL) commands to run programs on a set of nominated CICS regions. The following is a summary of these 3 coordinating programs:

#### **DFH0BAT1**

This sample program coordinates the disabling of a set of nominated transactions. This prevents the creation of new retained locks.

#### **DFH0BAT2**

This sample program coordinates the identification of retained lock information for a set of nominated data sets:

- For each data set, it issues a SET DSNAME RETRY command to try to resolve any retained locks that are due to transient failures, or failures that have been corrected.
- After a timed delay to allow retries to run, it issues an INQUIRE UOWDSNFAIL command to obtain information about any remaining shunted UOWs that have made uncommitted changes to the data set. It displays the information returned by the command, together with recommended procedures for resolving the locks.

#### **DFH0BAT3**

This sample program coordinates the forcing of locks for a set of nominated data sets:

- For each data set, it forces backout for the shunted in-doubt UOWs
- After a timed delay to allow the forced backouts to run, it resets the locks for any commit-failed or backout-failed UOWs.

The DFH0BAT3 sample program is also useful for resolving pending backouts after a failure to forward recover a data set.

The components used by the three coordinating programs is summarized in the following table.

*Table 11. Summary of the components used by the DFH0BATx sample programs*

Stage	Tranid	Initial program	DPL programs	TD queues	Mapset
Disable	BAT1	DFH0BAT1	DFH0BAT4	BATA BATX	DFH0BM1
Identify	BAT2	DFH0BAT2	DFH0BAT5 DFH0BAT7	BATA BATD	DFH0BM2
Force	BAT3	DFH0BAT3	DFH0BAT6 DFH0BAT8	BATA BATD	DFH0BM3

The programs are written VS COBOL II only, and are supplied with the necessary BMS maps and other copybooks. A summary of the processing performed by each program is given in the following table:

Table 12. Functional summary of the DFH0BATx programs

Program	Functional overview
DFH0BAT1	<p>DFH0BAT1 is invoked by transaction BAT1 in the CICS region selected as the coordinator, controlling the disabling of specified transactions. Reads 2 extrapartition TD queues: (1) BATX for the ids of transactions to be disabled, and (2) BATA for the applids of the target CICS regions.</p> <p>Issues DPL requests to DFH0BAT4 in each of the target regions to disable the named transactions. Any errors returned by each DFH0BAT4 are displayed using BMS map DFH0BM1.</p>
DFH0BAT2	<p>DFH0BAT2 is invoked by transaction BAT2 in the CICS region selected as the coordinator, controlling the gathering of retained lock information for the specified data sets. Reads 2 extrapartition TD queues: (1) BATD for the names of data sets, and (2) BATA for the applids of the target CICS regions.</p> <p>For each data set, DFH0BAT2 issues a DPL request to DFH0BAT7, in each target CICS region, to retry backout failures associated with the data set. When the DPL requests to DFH0BAT7 for a data set are completed, and after a timed delay, the program issues DPL requests to DFH0BAT5 to gather retained lock information from each of the target CICS regions for the same data set. The retained lock information from DFH0BAT5 invocations is received in a temporary storage queue (DFH0BQ2) and is displayed using BMS mapset DFH0BM2.</p> <p>This process of issuing DPL requests to DFH0BAT7 and DFH0BAT5 is repeated for each of the data set names obtained from BATD.</p>
DFH0BAT3	<p>DFH0BAT3 is invoked by transaction BAT3 in the CICS region chosen to be the coordinator region. It initiates the forced backout of any in-doubt units of work, and the forced release of retained locks, for specified data sets. Reads 2 extrapartition TD queues: (1) BATD for the names of data sets, and (2) BATA for the applids of the target CICS regions.</p> <p>For each data set, DFH0BAT3 issues a DPL request to DFH0BAT6, in each target CICS region, to force the backout of in-doubt units of work associated with the data set. When the DPL requests to DFH0BAT6 for a data set are completed, and after a timed delay, the program issues DPL requests to DFH0BAT8 to force the release of retained locks in each target region for the same data set. Messages from the DFH0BAT6 and DFH0BAT8 invocations are displayed using BMS mapset DFH0BM3.</p> <p>This process of issuing DPL requests to DFH0BAT6 and DFH0BAT8 is repeated for each of the data set names obtained from BATD.</p>
DFH0BAT4	<p>Linked by DPL request from DFH0BAT1 to disable specified transactions.</p>
DFH0BAT5	<p>Linked by DPL request from DFH0BAT2 to gather and return retained lock information to its caller.</p>
DFH0BAT6	<p>Linked by DPL request from DFH0BAT3 to force the backout of in-doubt units of work.</p>

Table 12. Functional summary of the DFH0BATx programs (continued)

Program	Functional overview
DFH0BAT7	Linked by DPL request from DFH0BAT3 to retry any backout failures.
DFH0BAT8	Linked by DPL request from DFH0BAT3 to force the release of retained locks.

For more information about the sample programs, see the comments in the prolog of each of the programs.

## Installing the sample programs

The resource definitions for the sample programs are supplied in the CSD in group DFH\$BAT. If you are able to use the definitions unmodified, add this group to one of the group lists you use at CICS startup on a cold start, or install the group while CICS is running using the CEDA install command. If you want to modify the resource definitions (to specify different DDNAMEs for the TD queues, for example), copy the group into another group and make the required changes. Add your copied group name to a group list, or install using CEDA.

## Preparing input for the sample programs

The three coordinating programs require input from extrapartition transient data queues. These TD queues provide the parameters the sample programs need. The TD queues and the parameters they hold are:

**BATA** The applids of the CICS regions involved in the quiesce operation

**BATX** The transaction ids of any transactions that are to be disabled

**BATD** The data set names that are to be quiesced

To prepare these TD queues and the control information:

- Define the sequential data sets for the TD queues as fixed block data sets with an 80 byte block size. You can define these either in the CSD (the preferred method) or in the DCT.
  - If you define the queues in the CSD, specify the data set names for dynamic allocation, and you do not need DD statements in the startup JCL. As a consequence of dynamic allocation, when a TD queue is closed, the underlying data set is de-allocated, which means that it can then be modified by, say, a TSO editor. This means that data sets can be modified without having to bring CICS down, which is not possible with the DCT where DD statements are required in the CICS startup JCL.
  - If you define the queues in the DCT, you must include the necessary DD statements for ddnames BATA, BATX, and BATD.
- Dynamic allocation allows you to use the TSO editor to enter the data into the data sets before they are dynamically allocated when the TD queues are opened.

**Note:** These definitions and TD queues need only be available to the CICS region you select to be the coordinator. They do not need to be defined to the target CICS regions. The queue names are coded in the programs, but you can change these if you want to use names that conform to your own naming conventions.

---

## Chapter 17. 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=CICSTS13.CICS.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 193.

### 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 37 is an example of a summary report produced by DFHMSCAN.

DFHMSCAN PROGRAM - SUMMARY LISTING							
MODULE	SIZE	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.		NUMBER	ASSEMBLER	COBOL	PL/I	ADDRESS	
MODULES		OF MACRO	MACRO	MACRO	MACRO	CSA	
SCANNED		PROGRAMS	PROGRAMS	PROGRAMS	PROGRAMS	PROGRAMS	
	4	2	1	1	0	1	

Figure 37. 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.

- An 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 18. Sign-on 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 Transaction Server for OS/390, because CICS Transaction Server for OS/390 does not support generation of SNTs.

The DFHSNMIG program is link-edited as APF-authorized and is installed in the CICSTS13.CICS.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—CICS SNT to 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 196.
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 196.

**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

Figure 38 shows an example job stream to run the DFHSNMIG program. The job

```
//SNTMIG JOB 'accounting information',  
//          CLASS=A,USER=userid,PASSWORD=password  
//SNTMIGR EXEC PGM=DFHSNMIG  
//STEPLIB DD DSN=CICSTS13.CICS.SDFHAUTH,DISP=SHR  
//DFHSNT DD DSN=CIC330.SDFHAUTH,DISP=SHR  
//CLIST DD DSN=CICSTS13.CICS.MIGRATE.CLIST,DISP=(,CATLG),  
//          SPACE=(TRK,(10,10),RLSE),UNIT=SYSDA,  
//          RECFM=VB,LRECL=256,BLKSIZE=4096  
//SYSPRINT DD SYSOUT=A  
//*
```

Figure 38. Sample job stream to run the DFHSNMIG program

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 Transaction Server for OS/390.

**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  
//SYSPRT DD SYSOUT=A  
//SYSUDUMP DD SYSOUT=A  
//SYSTSIN DD *  
EXEC 'CICSTS13.CICS.MIGRATE.CLIST' LIST  
/*  
//
```

Figure 39. Batch job to execute the CLIST created by the DFHSNMIG program

---

## Chapter 19. 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 20. Message editing utility (DFHMEU)

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 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 200.

**Note:** Several of the CICS messages cannot be changed by the message editing utility. In the *CICS Messages and Codes*, these messages are annotated accordingly.

---

### Installing the message editing utility

The library data sets and modules needed by the message editing utility are installed as part of the CICS Transaction Server for OS/390 Release 3 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 200.

#### Utility data sets

The following partitioned data sets are used by the message editing utility.

##### Message source data set

This data set, CICSTS13.CICS.SDFHMSRC, contains message source files for all languages serviced by IBM.

##### Executable files (CLISTs) data set

This data set, CICSTS13.CICS.SDFHCLIB, contains the message editing utility executable CLIST.

##### Load library

This data set, CICSTS13.CICS.SDFHLLIB, contains the load modules for the message editing utility.

##### Messages library

This data set, CICSTS13.CICS.SDFHMLIB, contains the modules for messages internal to the message editing utility.

### ISPF panel library

This data set, CICSTS13.CICS.SDFHPLIB, contains the panels for the message editing utility.

### Internal tables library

This data set, CICSTS13.CICS.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, CICSTS13.CICS.SDFHLANG, contains the table of all valid language codes supported by CICS.

These data sets, except CICSTS13.CICS.SDFHTLIB, are created automatically when you install CICS Transaction Server for OS/390 Release 3. The CICSTS13.CICS.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 CICSTS13.CICS.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(CICS530)
```

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 'CICSTS13.CICS.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 page
1	Start the message editing utility	201
2	Specify your message editing utility default values (the first time run)	201
3	Perform actions on message data sets (from main panel) such as: <ul style="list-style-type: none"><li>• Copy message data set members</li><li>• Select message sets to be edited</li><li>• Edit selected messages</li><li>• Assemble and link-edit changed message set members</li><li>• Generate a message load module</li></ul>	205

Step	Task	For details, see page
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.	209
5	(If needed) Apply PTFs	210

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 ISPCMD table. For example:

```
MEU 3 SELECT CMD(EXEC 'CICSTS13.CICS.SDFHCLIB(DFHMEUCL)'
    'MEULIB(CICSTS13.CICS)')
```

- Execute as a TSO function from the ISPF command line. For example:

```
TSO EX 'CICSTS13.CICS.SDFHCLIB(DFHMEUCL)' 'MEULIB(CICSTS13.CICS)'
```

**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 40 on page 202.

**Note:** When you first start the message editing utility, the following message is overlaid on the CICS macro library and 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 41 on page 203.

When you have defined your default values, any subsequent start of the message editing utility displays the message editing utility *Main* panel first.

```

Message Editing Utility - Set defaults          Page 1 of 2
More: +

Values are saved when ENTER, Forward (F8), or End (F3) is pressed.

MEU target data sets index  userid_____

Current language (NATLANG)  A + (ENG) Alternative English

Replace members during
English source copy? . NO_ Yes/No

SMP/E maintained SDFHMSRC . CICSTS13.CICS.SDFHMSRC_____

DFHMEUL load library . . . userid.load.library_____

CICS macro library . . . . CICSTS13.CICS.SDFHMAC_____

CICS LOAD library . . . . CICSTS13.CICS.SDFHAUTH_____
COMMAND ==>>> _____
F1=Help      F2=Split    F3=End      F4=Language  F5=Refresh  F8=Forward
F9=Swap      F12=Cancel

```

Figure 40. Message editing utility set defaults panel (1 of 2)

**Note:** After the *MEU target data sets index* and the *SMP/E maintained SDFHMSRC* parameters have been set they should not be changed. The message editing utility creates a copy of the *SMP/E maintained SDFHMSRC* for its own use, called *MEU target data sets index.SDFHMSRC*. 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 42 on page 205.) 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.)

The 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 . . //*_____

COMMAND ==> _____
F1=Help      F2=Split      F3=End      F5=Refresh  F7=Backward F9=Swap
F12=Cancel
```

Figure 41. 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.

### 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 42 on page 205. The languages supported by the message editing utility are listed in the *Language selection panel* and for reference in Table 13 on page 204.

Table 13. 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
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.
  - C - Simplified Chinese 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
           Copied  ENG  Alternative English
-   A
-   B          PTB  Brazilian Portuguese
-   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 42. 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 43 on page 206.

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      .         .           .           .       .       .
-      DFHMEDEE      .         .           .           .       .       .
-      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 43. 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 40 on page 202.)

The *PTF update* panel can be accessed by ApplyPTF (F10). (See Figure 46 on page 210.)

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.SDFHMSRC

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.

**Note:**

When editing RP messages, these messages are split between four message sets as follows:

- DFHMER0x 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 FC messages, these messages are split between four message sets as follows:

- DFHMEFAx for message numbers from 0000 to 0499
- DFHMEFBx for message numbers from 5000 to 2000
- DFHMEFCx for message numbers from 2001 to 6999
- DFHMEFDx for message numbers from 7000 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 Messages and Codes*. An example of the *Message number selection* panel is shown in Figure 44 on page 208.

```

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 44. 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 45) 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 45.

```

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 45. Example of message editing utility edit message panel

For information and rules about editing CICS messages, see “Rules for editing and translating” on page 212.

**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 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 46. To apply a PTF:
  - a. Complete the data set details for the SMP/E maintained SDFHMSRC 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 SDFHMSRC. CICSTS13.CICS.SDFHMSRC _____
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 46. 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 47 on page 212 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.SDFHMSRC

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.SDFHMSRC

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 47. 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 ==> _____ 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 48. Message editing utility edit message panel showing types of inserts

In the example in Figure 48, 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 49 on page 214. 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 49. 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 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 50.

```

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 50. 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, 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
TRANNUM	6 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 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 51. Message editing utility help contents panel*

To display help information for a specific topic, type the number for the topic, press ENTER.

---

## Chapter 21. Shutdown assist program (DFHCESD)

You can use the SDTRAN system initialization parameter, or the SDTRAN option of the PERFORM SHUTDOWN command, to specify the name of the *shutdown assist transaction* to be started during CICS shutdown processing. The default transaction is CESD, which invokes the CICS-supplied shutdown assist program, DFHCESD. You can use the supplied program “as is”, or as a sample on which to base your own shutdown transaction.

Unless SDTRAN=NO was specified at system initialization, or NOSDTRAN on the PERFORM SHUTDOWN command, the shutdown assist transaction is started automatically at the beginning of a normal or immediate shutdown. It runs under the userid authority of the issuer of the shutdown command.

**Note:** If the program named by the shutdown transaction cannot be loaded, CICS waits indefinitely for all user tasks to complete, which may cause shutdown to hang. *This happens on an immediate, as well as on a normal, shutdown.*

You can use the shutdown assist transaction to help solve two of the problems that can arise when shutting down CICS:

- On a normal shutdown, CICS waits for all running tasks to finish before entering the second stage of shutdown. Long-running or conversational transactions can cause an unacceptable delay, or can require operator intervention.
- On an immediate shutdown, CICS does not allow running tasks to finish; and backout is not performed until emergency restart. This can cause an unacceptable number of units of work to be shunted, with a consequent retention of locks.

---

### Actions of the default program

The default shutdown assist program, DFHCESD, attempts to purge and backout long-running tasks. It ensures that as many tasks as possible commit or backout cleanly, enabling CICS to shut down in a controlled manner.

Tasks are purged in three steps; successive steps use increasingly stronger purge techniques and are invoked only if tasks refuse to disappear from the system. The three purge steps that DFHCESD moves through are:

1. Normal purge is issued for all remaining tasks.
2. VTAM is force closed and IRC is closed immediately.
3. CICS is shut down using PERFORM SHUT IMMEDIATE. (Note that this step does not cause the shutdown assist transaction to run again.)

To check whether tasks are ending sufficiently quickly, DFHCESD samples the number present in the system. It performs a purge operation, and moves on to the next step, only if the number of tasks does not reduce over eight samples (normal shutdown) or four samples (immediate shutdown). After taking a sample, DFHCESD issues a delayed EXEC CICS START request for itself, passing the current sample count in a temporary storage (TS) queue record. The new invocation of DFHCESD also takes a sample, and compares this with the last

sample from the TS queue record. It then decides whether to carry out the purge operation and move to the next step, or to remain on the current step.

The information passed to DFHCESD in the TS queue record is:

SDFN	Char(2)	Step to be performed (00,01,02,03)	SDXN	Char(4)
		Task number of task that started shutdown	SDET	Bin(15)
		Number of samples giving the value in SDNT	SDNT	Bin(31)
		of tasks in the system at last sample		Number

On the initial invocation, SDFN is '00', SDXN is set to the task number of the shutdown task, and SDNT and SDET are zero.

Thus, DFHCESD's processing sequence is as follows:

#### **Initial step 00**

In step 00, if shutdown is NORMAL, DFHCESD puts out a message and waits for two minutes. It then issues a delayed start of CESD every two seconds.

- Every two seconds, the number of transactions in the system is sampled.
- If the number is unchanged over four or eight samples (depending on whether this is a normal or immediate shutdown), the first of the purge steps is taken.

#### **Purge step 01**

The transaction dump dataset is closed and task purge is issued for all remaining transactions, with messages giving details of each transaction still running and each UOW shunted.

- Every two seconds, the number of transactions in the system is sampled.
- If the number is unchanged over four or eight samples, step 02 is taken.

#### **Purge step 02**

Unless VTAM persistent sessions support is being used (that is, unless the persistent session delay interval is set to a value greater than zero), VTAM is force closed and IRC is closed immediately.

- Every two seconds, the number of transactions in the system is sampled.
- If the number is unchanged over four or eight samples, step 03 is taken.

#### **Purge step 03**

CICS is shut down abnormally, with messages giving details of each transaction still running and each UOW shunted.

---

## **The sample programs**

Source code versions of the default program are provided in Assembler, COBOL, and PL/I, in the CICSTS13.CICS.SDFHSAMP library. They are named DFHCESD, DFH0CESD, and DFH\$CESD, respectively. All contain the same logic. Resource definitions of DFH0CESD and DFH\$CESD and their associated transactions, SDA1 and SDA2, are in the sample CSD group DFH\$SDAP. Definitions of DFHCESD and CESD are in the DFHSDAP CSD group (which is included in the default startup group list, DFHLIST).

If you use one of the samples as the basis for your own program:

- Modify the program. There are no restrictions on the EXEC CICS API and SPI commands you can use, except that EXEC CICS START can start only shutdown-enabled programs.
- If necessary, modify the supplied resource definitions in group DFH\$SDAP or DFHSDAP. (You may decide, for example, to name your transaction and program differently from the samples.) On the supplied transaction definitions, command and resource security (specified by the CMDSEC and RESSEC attributes) are not active.
- Ensure that group DFH\$SDAP or DFHSDAP, as appropriate, is included in your startup group list.
- Specify the name of your shutdown assist transaction on the SDTRAN system initialization parameter.

Figure 52 on page 220 shows some example messages generated by a run of the DFHCESD sample program.

```

16.14.08 JOB09120 +DFHTM1715 IY LX1 CICS/ESA is being quiesced by userid CICSUSER in transaction CEMT
                  at netname IGCS21F.
16.14.08 JOB09120 +DFHDM0102I IY LX1 CICS is quiescing.
16.14.08 JOB09120 +DFHTM1781 IY LX1 CICS shutdown cannot complete because some non-system user tasks
                  have not terminated.
16.14.09 JOB09120 +DFHCESD IY LX1 SHUTDOWN ASSIST TRANSACTION CESD STARTING. SHUTDOWN IS NORMAL.
16.14.09 JOB09120 +DFHCESD IY LX1 LIST OF SHUNTED UNITS OF WORK IN THE SYSTEM FOLLOWS.
16.14.09 JOB09120 +DFHCESD IY LX1 SHUNTED TRANSACTION RFI4, TERMID S21F, UNIT OF WORK ABE7194B52539603
16.16.09 JOB09120 +DFHCESD IY LX1 THERE ARE NOW 0100 TASKS STILL IN THE SYSTEM.
16.16.26 JOB09120 +DFHDU0303I IY LX1 Transaction Dump Data set DFHDMPA closed.
16.16.26 JOB09120 +DFHCESD IY LX1 PURGING TRANID KK07, TERMID      , USERID CICSUSER, TASKNO 000113
16.16.26 JOB09120 +DFHCESD IY LX1 PURGING TRANID KK15, TERMID      , USERID CICSUSER, TASKNO 000119
16.16.26 JOB09120 +DFHCESD IY LX1 PURGING TRANID KK16, TERMID      , USERID CICSUSER, TASKNO 000120
16.16.26 JOB09120 +DFHCESD IY LX1 PURGING TRANID KK23, TERMID      , USERID CICSUSER, TASKNO 000127
16.16.26 JOB09120 +DFHCESD IY LX1 PURGING TRANID KK24, TERMID      , USERID CICSUSER, TASKNO 000128
16.16.26 JOB09120 +DFHCESD IY LX1 PURGING TRANID KK29, TERMID      , USERID CICSUSER, TASKNO 000133
16.16.26 JOB09120 +DFHCESD IY LX1 PURGING TRANID KK36, TERMID      , USERID CICSUSER, TASKNO 000139
16.16.26 JOB09120 +DFHCESD IY LX1 PURGING TRANID KK38, TERMID      , USERID CICSUSER, TASKNO 000140
16.16.26 JOB09120 +DFHCESD IY LX1 PURGING TRANID KK47, TERMID      , USERID CICSUSER, TASKNO 000144
16.16.26 JOB09120 +DFHCESD IY LX1 PURGING TRANID SK07, TERMID      , USERID CICSUSER, TASKNO 000161
16.16.26 JOB09120 +DFHCESD IY LX1 PURGING TRANID SK15, TERMID      , USERID CICSUSER, TASKNO 000167
16.16.26 JOB09120 +DFHCESD IY LX1 PURGING TRANID SK16, TERMID      , USERID CICSUSER, TASKNO 000168
16.16.26 JOB09120 +DFHCESD IY LX1 PURGING TRANID SK23, TERMID      , USERID CICSUSER, TASKNO 000175
16.16.26 JOB09120 +DFHCESD IY LX1 PURGING TRANID SK24, TERMID      , USERID CICSUSER, TASKNO 000176
16.16.26 JOB09120 +DFHCESD IY LX1 PURGING TRANID SK29, TERMID      , USERID CICSUSER, TASKNO 000181
16.16.26 JOB09120 +DFHCESD IY LX1 PURGING TRANID SK36, TERMID      , USERID CICSUSER, TASKNO 000187
16.16.26 JOB09120 +DFHCESD IY LX1 PURGING TRANID SK38, TERMID      , USERID CICSUSER, TASKNO 000188
16.16.26 JOB09120 +DFHCESD IY LX1 PURGING TRANID SK44, TERMID      , USERID CICSUSER, TASKNO 000189
16.16.26 JOB09120 +DFHCESD IY LX1 PURGING TRANID SK47, TERMID      , USERID CICSUSER, TASKNO 000192
16.16.26 JOB09120 +DFHCESD IY LX1 PURGING TRANID SK49, TERMID      , USERID CICSUSER, TASKNO 000194
16.16.26 JOB09120 +DFHCESD IY LX1 PURGING TRANID RSD1, TERMID S234, USERID CICSUSER, TASKNO 000418
16.16.26 JOB09120 +DFHCESD IY LX1 PURGING TRANID SK49, TERMID      , USERID CICSUSER, TASKNO 000424
16.16.26 JOB09120 +DFHCESD IY LX1 PURGING TRANID SK47, TERMID      , USERID CICSUSER, TASKNO 000426
16.16.26 JOB09120 +DFHCESD IY LX1 PURGING TRANID SK44, TERMID      , USERID CICSUSER, TASKNO 000429
16.16.26 JOB09120 +DFHCESD IY LX1 PURGING TRANID SK23, TERMID      , USERID CICSUSER, TASKNO 000437
16.16.26 JOB09120 +DFHCESD IY LX1 PURGING TRANID SK16, TERMID      , USERID CICSUSER, TASKNO 000444
16.16.26 JOB09120 +DFHCESD IY LX1 PURGING TRANID RFS1, TERMID X134, USERID CICSUSER, TASKNO 000531
16.16.26 JOB09120 +DFHCESD IY LX1 PURGING TRANID RFS1, TERMID X150, USERID CICSUSER, TASKNO 000532
16.16.26 JOB09120 +DFHCESD IY LX1 PURGING TRANID RFS1, TERMID X141, USERID CICSUSER, TASKNO 000533
16.16.26 JOB09120 +DFHCESD IY LX1 PURGING TRANID RFS1, TERMID X124, USERID CICSUSER, TASKNO 000534
16.16.26 JOB09120 +DFHCESD IY LX1 PURGING TRANID RFS1, TERMID X147, USERID CICSUSER, TASKNO 000535
16.16.26 JOB09120 +DFHCESD IY LX1 PURGING TRANID RFS1, TERMID X152, USERID CICSUSER, TASKNO 000536

```

Figure 52. Example messages generated by the sample DFHCESD program (Part 1 of 3)



```

16.16.26 JOB09120 +DFHCESD IY LX1 PURGING TRANID RFS1, TERMID X167, USERID CICSUSER, TASKNO 000605
16.16.26 JOB09120 +DFHCESD IY LX1 PURGING TRANID RFS1, TERMID X138, USERID CICSUSER, TASKNO 000606
16.16.26 JOB09120 +DFHCESD IY LX1 PURGING TRANID RFS1, TERMID X161, USERID CICSUSER, TASKNO 000607
16.16.26 JOB09120 +DFHCESD IY LX1 PURGING TRANID RFS1, TERMID X146, USERID CICSUSER, TASKNO 000608
16.16.26 JOB09120 +DFHCESD IY LX1 PURGING TRANID RFS1, TERMID X140, USERID CICSUSER, TASKNO 000609
16.16.26 JOB09120 +DFHCESD IY LX1 PURGING TRANID SK52, TERMID , USERID CICSUSER, TASKNO 000613
16.16.27 JOB09120 +DFHCESD IY LX1 LIST OF SHUNTED UNITS OF WORK IN THE SYSTEM FOLLOWS.
16.16.27 JOB09120 +DFHCESD IY LX1 SHUNTED TRANSACTION RFI4, TERMID S21F, UNIT OF WORK ABE7194B52539603
16.16.28 JOB09120 +DFHCESD IY LX1 THERE ARE NOW 0081 TASKS STILL IN THE SYSTEM.
16.16.31 JOB09120 +DFHCESD IY LX1 THERE ARE NOW 0073 TASKS STILL IN THE SYSTEM.
16.16.43 JOB09120 +DFHCESD IY LX1 THERE ARE NOW 0072 TASKS STILL IN THE SYSTEM.
16.16.45 JOB09120 +DFHCESD IY LX1 THERE ARE NOW 0070 TASKS STILL IN THE SYSTEM.
16.16.47 JOB09120 +DFHCESD IY LX1 THERE ARE NOW 0004 TASKS STILL IN THE SYSTEM.
16.17.05 JOB09120 +DFHCESD IY LX1 LIST OF SHUNTED UNITS OF WORK IN THE SYSTEM FOLLOWS.
16.17.05 JOB09120 +DFHCESD IY LX1 SHUNTED TRANSACTION RFI4, TERMID S21F, UNIT OF WORK ABE7194B52539603
16.17.07 JOB09120 +DFHZC2316 IY LX1 VTAM ACB is closed
16.17.09 JOB09120 +DFHCESD IY LX1 THERE ARE NOW 0002 TASKS STILL IN THE SYSTEM.
16.17.09 JOB09120 +DFHTM1782I IY LX1 All non-system tasks have been successfully terminated.
16.17.30 JOB09120 +DFHRM0131 IY LX1 Resynchronization required with IRC resources.
16.17.30 JOB09120 +DFHRM0131 IY LX1 Resynchronization required with LU62 resources.
16.17.30 JOB09120 +DFHRM0131 IY LX1 Resynchronization required with IND resources.
16.17.40 JOB09120 +DFHRM0203 IY LX1 There are 1 indoubt, 0 commit-failed and 0 backout-failed UOWs.
16.17.56 JOB09120 +DFHRM0130 IY LX1 Recovery manager has successfully quiesced.
16.18.01 JOB09120 +DFHKE1799 IY LX1 TERMINATION OF CICS/ESA IS COMPLETE.

```

Figure 52. Example messages generated by the sample DFHCESD program (Part 3 of 3)

---

## Chapter 22. Recovery manager utility program (DFHRMUTL)

This chapter describes the recovery manager utility program, DFHRMUTL, for overriding the type of CICS startup produced by a START=AUTO, and improving the performance of cold and initial starts.

**Note:** For detailed information about the type of startup produced by each possible combination of START setting, global catalog and system log contents, and autostart override, see the *CICS System Definition Guide*.

---

### Overview of DFHRMUTL

DFHRMUTL processes the global catalog data set. It can insert or modify the recovery manager autostart override record. Optionally, it can extract a subset of the catalog records to build a reduced new catalog for a cold start.

You can use the recovery manager utility program to:

- Set or reset the recovery manager autostart override record on the global catalog.
- Examine the setting of the autostart override record on the global catalog.
- Copy that part of the catalog needed for a cold start to a new global catalog.

If a new catalog is built using DFHRMUTL, CICS is able to perform only a cold start or an initial start with the new catalog. The performance of these starts will, however, be better than that of a cold or initial start with a full catalog.

DFHRMUTL sets a return code indicating if it has succeeded.

### Input to DFHRMUTL

You can specify what you want the utility to do by supplying parameters in a single optional record in the input data set, SYSIN. See “Specifying parameters” on page 224.

You may need to supply one or two CICS global catalog data sets:

#### DFHGCD

The catalog from which a copy is extracted or, if no copy is being made, the one in which the autostart override record is placed.

#### NEWGCD

The catalog which is cleared and receives the copy, if one is requested.

### Output from DFHRMUTL

DFHRMUTL writes some or all of the following to the output data set, SYSPRINT:

- The input record from SYSIN.
- Error messages.
- A summary of the autostart override record found on the global catalog, DFHGCD.
- Whether or not the DFHGCD catalog is a reduced copy from a previous run of DFHRMUTL.
- The autostart override record that has been set on DFHGCD, or on NEWGCD if a copy is made.

The catalogs DFHGCD and NEWGCD may be updated. If no copy is requested, DFHGCD may have an override record inserted, or updated. If a copy is requested, DFHGCD is unchanged, NEWGCD is cleared, and the copy and new override record written to NEWGCD.

---

## JCL requirements

DFHRMUTL runs as a standard operating system job. You require a JOB statement, an EXEC statement, and DD statements defining input and output. "Examples of using DFHRMUTL" on page 226 contains some example jobs that illustrate the uses of DFHRMUTL.

### DD statements

This section describes the DD statements for the input and output data sets used by DFHRMUTL.

#### STEPLIB DD

Defines a partitioned data set (DSORG=PO) containing DFHRMUTL. If DFHRMUTL is in the link list, this statement is not required.

#### SYSPRINT DD

Defines the output data set for results, information and error messages. The DCB parameters for this data set are RECFM=FBA and LRECL=133.

The block size can be provided on the SYSPRINT DD statement and must be a multiple of 133. The default is 133.

#### SYSIN DD

Defines the input data set. This file must be in 80-byte record format.

#### DFHGCD DD

Defines the input global catalog data set, which may be empty. This catalog may be updated unless the COLD\_COPY parameter is specified, in which case it is only read.

**Note:** An empty catalog data set, after having an override record inserted by DFHRMUTL, may then be used by a CICS system for startup.

#### NEWGCD DD

Defines the output global catalog data set. This statement is not required unless the COLD\_COPY parameter is specified. If COLD\_COPY is specified the NEWGCD data set is first cleared and then has DFHGCD records and an override record added to it. It must have been defined with the VSAM REUSE attribute.

---

## Specifying parameters

You can use the parameters SET\_AUTO\_START and COLD\_COPY to control the actions that DFHRMUTL takes.

The first record of SYSIN must contain all of the parameters for the utility. If the SYSIN data set is empty, DFHRMUTL outputs a summary of the autostart override record it finds on the DFHGCD catalog. If SYSIN contains more than one record, it is an error.

If specified, the parameters must be separated by commas and contain no embedded blanks. After the parameters, which must be the first non-blank characters of the record, all other characters are ignored.

## SET\_AUTO\_START={AUTOASIS|AUTOCOLD|AUTODIAG|AUTOINIT}

The type of the next startup, if the START=AUTO system initialization parameter is specified. The start-type is placed in the autostart override record of either DFHGCD or (if COLD\_COPY is specified) NEWGCD.

### AUTOASIS

Perform the default startup, either warm or emergency. If START=AUTO is used on the next startup, the start-type is based on the recovery manager control record. This is the startup you would get if the autostart override record were not present.

This value is not compatible with the COLD\_COPY keyword, nor is it allowed if the catalog you are updating is the result of a COLD\_COPY and CICS has not used the catalog since then. Either of these combinations would result in CICS performing a warm or emergency restart with insufficient information in the global catalog.

### AUTOCOLD

Perform a cold start. If START=AUTO is used on the next startup, CICS performs a cold start, if a cold start is possible at that time.

If the input catalog is empty (or has insufficient information in it) AUTOCOLD is not allowed.

The following parameters, unless overridden at systems initialization, are taken from the catalog and are ignored by SET\_AUTO\_START=AUTOCOLD.

- ICV
- ICVTSD
- MAXOPENTCBS
- MROBTCH
- PRTYAGE

### AUTODIAG

Perform a diagnostic run. If START=AUTO is used on the next startup, CICS performs a diagnostic run. On a diagnostic run, CICS:

1. Produces a dump of the CICS system state, retrieved from the CICS system log.
2. Terminates. Note that, on a diagnostic run, CICS *performs no recovery work and no new work*.

A diagnostic run is used to diagnose problems on the CICS system log. The output produced by a diagnostic run is usually passed to IBM Service.

If the system log becomes corrupt, CICS sets the recovery manager autostart override record in the global catalog so that the next automatic start (START=AUTO) is a diagnostic run. However, there may be other occasions (when perhaps the system log is still readable) when you feel it would be useful to perform a diagnostic run. The AUTODIAG option enables you to specify a diagnostic run manually.

For detailed information about diagnostic runs, and the circumstances in which you might need to perform one, see the *CICS Problem Determination Guide*.

#### Notes:

1. Unlike the other options of SET\_AUTO\_START, whose effects are limited to the next CICS startup, AUTODIAG has a persistent effect.

That is, the autostart override record of DFHGCD is set to produce a diagnostic run on each subsequent AUTO start, and can only be reset by running DFHRMUTL again.

2. AUTODIAG is not compatible with the COLD\_COPY keyword.

#### **AUTOINIT**

Perform an initial start. If START=AUTO is used on the next startup, CICS performs an initial start, whatever the other contents of the global catalog.

The following parameters, unless overridden at systems initialization, are taken from the catalog and are ignored by SET\_AUTO\_START=AUTOINIT.

- ICV
- ICVTSD
- MAXOPENTCBS
- MROBTCH
- PRTYAGE

#### **COLD\_COPY**

Make a reduced copy of DFHGCD in NEWGCD. Create in NEWGCD a copy of only those records from DFHGCD that CICS needs to perform a cold start, and update NEWGCD with the autostart override record specified by the SET\_AUTO\_START parameter.

#### **Notes:**

1. All changes caused by SET\_AUTO\_START are made to the NEWGCD data set, and DFHGCD is not changed.
2. COLD\_COPY is incompatible with the AUTOASIS and AUTODIAG options of SET\_AUTO\_START. If you specify COLD\_COPY and either of these values of SET\_AUTO\_START, it is an error.

---

## **Return codes**

DFHRMUTL sets one of the following return codes:

- |           |  |
|-----------|--|
| <b>00</b> | The parameters are valid and all reads and writes to the input and output data sets were successful. |
| <b>16</b> | One or more errors were detected during execution. An error message is output.                       |

Errors that DFHRMUTL may detect are:

- A read or write error for the SYSIN or SYSPRINT data set
- A read or write error for one of the catalog data sets
- A syntax error in the parameters
- A parameter that is incompatible with the input catalog data set
- An invalid combination of parameters.

---

## **Examples of using DFHRMUTL**

The following sections illustrate the uses of DFHRMUTL.

### **Setting an initial start without operator intervention**

Figure 53 on page 227 shows the statements required to update the global catalog so that CICS performs an initial start if START=AUTO is specified.

You could use this job to modify a newly-defined global catalog. This would mean that could retain START=AUTO for all your CICS start jobs, including the first with a new global catalog.

**Note:** If you use this step to initialize a newly-defined global catalog, you should use the DFHCCUTL utility to initialize the local catalog too. (If you use it to reinitialize an existing global catalog, it is not necessary to initialize the local catalog.) For information about initializing catalog data sets, see the *CICS System Definition Guide*.

```
//RMUTL EXEC PGM=DFHRMUTL,REGION=1M
//STEPLIB DD DSNAME=link.dataset,DISP=SHR
//SYSPRINT DD SYSOUT=A
//DFHGCD DD DSNAME=catalog.dataset,DISP=OLD
//SYSIN DD *
        SET_AUTO_START=AUTOINIT
/*
```

Figure 53. DFHRMUTL—setting the next auto start to be an initial start.

## Examining the override record

Figure 54 shows the statements required to examine the autostart override record on a CICS global catalog data set.

```
//RMUTL EXEC PGM=DFHRMUTL,REGION=1M
//STEPLIB DD DSNAME=link.dataset,DISP=SHR
//SYSPRINT DD SYSOUT=A
//DFHGCD DD DSNAME=catalog.dataset,DISP=OLD
//SYSIN DD *
/*
```

Figure 54. DFHRMUTL—examining an autostart override record

This JCL also reveals:

- If this is a catalog data set produced by a COLD\_COPY and not yet used by CICS (and so not populated with other records)
- If it is an “empty” catalog—that is, it does not contain a recovery manager control record.

## Resetting a warm or emergency start

Figure 55 shows the statements required to update the global catalog so that CICS performs a warm or emergency start if START=AUTO is specified. This enables you to undo the effects of a previous run of DFHRMUTL that set the autostart override record to AUTOINIT or AUTOCOLD. (The AUTOASIS override record is equivalent to there being no override record in the global catalog.)

If the global catalog data set was produced by a COLD\_COPY, or if it is empty, DFHRMUTL rejects the AUTOASIS value.

```
//RMUTL EXEC PGM=DFHRMUTL,REGION=1M
//STEPLIB DD DSNAME=link.dataset,DISP=SHR
//SYSPRINT DD SYSOUT=A
//DFHGCD DD DSNAME=catalog.dataset,DISP=OLD
//SYSIN DD *
        SET_AUTO_START=AUTOASIS
/*
```

Figure 55. DFHRMUTL—resetting a warm or emergency start

## Improving the performance of a cold start

Figure 56 shows the statements required to:

- Create a new global catalog data set consisting only of those records required for a cold start.
- Set the autostart override record of the new catalog to indicate a cold start.
- Replace the original catalog with the new one, if the creation step succeeded.

Because the original catalog data set is overwritten by a COLD\_COPY, it is not suitable for a warm or emergency start. DFHRMUTL does not allow you to reset the override record to read AUTOASIS.

```
//RMUTL EXEC PGM=DFHRMUTL,REGION=1M
//STEPLIB DD DSNAME=link.dataset,DISP=SHR
//SYSPRINT DD SYSOUT=A
//DFHGCD DD DSNAME=catalog.dataset,DISP=OLD
//NEWGCD DD DSNAME=newcatalog.dataset,DISP=OLD
//SYSIN DD *
        SET_AUTO_START=AUTOCOLD,COLD_COPY
/*
//          IF (RMUTL.RC=0) THEN
// * Step to be performed if RMUTL succeeds
//COPY EXEC PGM=IDCAMS
//DFHGCD DD DSNAME=catalog.dataset,DISP=OLD
//SYSPRINT DD SYSOUT=A
//NEWGCD DD DSNAME=newcatalog.dataset,DISP=OLD
//SYSIN DD *
        REPRO INFILE(NEWGCD) OUTFILE(DFHGCD) REUSE
/*
// * End of step
//          ENDIF
```

Figure 56. DFHRMUTL—setting the global catalog for a cold start. COLD\_COPY is used to improve performance.

---

## Chapter 23. BMS macro generation utility program (DFHBMSUP)

This chapter describes the BMS macro generation utility, DFHBMSUP, to recreate BMS macro statements from a mapset load module.

---

### Overview

DFHBMSUP can recreate the original BMS macros that were assembled to produce a mapset load module, when the macro statements are no longer available.

The utility program generates map definition macros that are equivalent to the originals, and thus can be used to recreate symbolic maps if the original source has been lost. However, it is not possible to recover the original field names used. Field names are generated by the utility and you can then edit them.

DFHBMSUP sets a return code indicating success or failure.

**Note:** DFHBMSUP cannot process mapset load modules created on CICS/OS/VS 1.7 and earlier releases.

### Input

All input information is defined in the JCL.

DFHBMSUP requires the following inputs:

**Input MAPSET**

Name defined in the PARM field of the EXEC JCL statement.

**Input MAPSET library**

Name defined in the DFHRPL DD statement.

### Output

DFHBMSUP provides the following outputs:

**Output map**

Name defined in the BMSOUT DD statement.

**Output map library**

Name defined in the BMSOUT DD statement.

### DD statements

This section describes the DD statements for the input and output data sets used by DFHBMSUP.

**STEPLIB DD**

Defines a partitioned data set (DSORG=PO) containing DFHBMSUP. If DFHBMSUP is in the link list, this statement is not required.

**DFHRPL DD**

Defines a partitioned data set (DSORG=PO) containing the mapset load module to be processed. The member name is supplied in the PARM field of the EXEC statement.

## BMSOUT DD

Defines a partitioned data set (DSORG=PO) to contain the BMS macro statements generated by the utility.

## Return codes

DFHBMSUP sets one of the following return codes:

- 0 Utility executed successfully.
- 4 Input mapset could not be located.
- 8 Output mapset could not be opened.

## Example of using DFHBMSUP

Figure 57 shows the statements required to process a BMS mapset load module, **BMSET01**, which is in the **INPUT.BMSLIB** library. Macro statements are generated and written to the **MAPOUT** member of the **OUTPUT.MACLIB** library.

```
//*****  
//* RUN THE DFHBMSUP PROGRAM *  
//* INPUT BMSET01 *  
//* OUTPUT MAPOUT *  
//* *  
//*****  
//*  
//RUNPROG EXEC PGM=DFHBMSUP,PARM='BMSET01',REGION=2M  
//STEPLIB DD DSN=CICSTS13.CICS.SDFHLOAD,DISP=SHR  
//BMSOUT DD DSN=OUTPUT.MACLIB(MAPOUT),  
// DISP=SHR  
//DFHRPL DD DSN=INPUT.BMSLIB,DISP=SHR  
//SYSUDUMP DD SYSOUT=*  
//*
```

Figure 57. DFHBMSUP—generating BMS macro statements.

## Example of DFHBMSUP output

The following macro statements were generated from the mapset load module, **BMSET40**.

```
# Note that the utility uses the name of the load module being disassembled as the  
# label on the DFHMSD macro, so that you can use it to build HTML template  
# datasets which are accessible to the original 3270 applications. If you specify the  
# suffixed version of the load module for a mapset as input to DFHBMSUP and you  
# intend to use the output source to build HTML for the mapset, you may need to  
# edit the DFHMSD label to remove the suffix.
```

```
# Also note that DFHBMSUP does not know whether STORAGE=AUTO was defined  
# in the original macro statements, so the DFHBMSUP output does not include  
# STORAGE=AUTO on the DFHMSD mapset definition macro. If you require  
# STORAGE=AUTO, edit the DFHMSD macro to include it.
```

You can edit all the names in the DFHBMSUP output to be more meaningful for your application.

```
* This is an unaligned mapset  
*  
TITLE 'BMSET40 Mapset MACRO Definition Listing'  
BMSET40 DFHMSD TYPE=DSECT,LANG=ASM,MODE=INOUT  
*
```

```

BMAP400 DFHMDI SIZE=(1,80),CTRL=(FRSET,FREEKB),COLUMN=1,LINE=1,      *
        MAPATTS=(COLOR,HILIGHT)
        DFHMDF POS=0,LENGTH=4,ATTRB=(ASKIP,BRT),COLOR=PINK,          *
        HILIGHT=REVERSE,INITIAL='BM40'
        DFHMDF POS=5,LENGTH=1,COLOR=BLUE
FLD00001 DFHMDF POS=16,LENGTH=45,ATTRB=(ASKIP,BRT),COLOR=NEUTRAL
        DFHMDF POS=62,LENGTH=1,COLOR=BLUE
FLD00002 DFHMDF POS=78,LENGTH=1,COLOR=YELLOW
BMAP401 DFHMDI SIZE=(9,80),CTRL=(FRSET,FREEKB),COLUMN=1,LINE=2,      *
        MAPATTS=(COLOR,HILIGHT)
        DFHMDF POS=0,LENGTH=1,COLOR=BLUE,INITIAL=' '
        DFHMDF POS=80,LENGTH=1,COLOR=BLUE,INITIAL=' '
        DFHMDF POS=160,LENGTH=1,COLOR=BLUE,INITIAL=' '
        DFHMDF POS=240,LENGTH=1,COLOR=BLUE,INITIAL=' '
        DFHMDF POS=320,LENGTH=1,COLOR=BLUE,INITIAL=' '
        DFHMDF POS=400,LENGTH=1,COLOR=BLUE,INITIAL=' '
        DFHMDF POS=480,LENGTH=1,COLOR=BLUE,INITIAL=' '
        DFHMDF POS=560,LENGTH=1,COLOR=BLUE,INITIAL=' '
        DFHMDF POS=658,LENGTH=39,COLOR=TURQUOISE,                    *
        INITIAL='THIS SHOULD BE IN THE MIDDLE OF LINE 10'
*
BMAP402 DFHMDI SIZE=(1,80),CTRL=(FRSET,FREEKB),COLUMN=1,LINE=11,      *
        MAPATTS=(COLOR,HILIGHT)
        DFHMDF POS=0,LENGTH=1,COLOR=BLUE,INITIAL=' '
BMAP403 DFHMDI SIZE=(1,80),CTRL=(FRSET,FREEKB),COLUMN=1,LINE=11,      *
        MAPATTS=(COLOR,HILIGHT)
        DFHMDF POS=17,LENGTH=41,COLOR=TURQUOISE,                      *
        INITIAL='THIS TEXT SHOULD NOT APPEAR ON THE SCREEN'
*
BMAP404 DFHMDI SIZE=(10,80),CTRL=(FRSET,FREEKB),COLUMN=1,LINE=12,      *
        MAPATTS=(COLOR,HILIGHT)
        DFHMDF POS=18,LENGTH=39,COLOR=TURQUOISE,                      *
        INITIAL='THIS SHOULD BE IN THE MIDDLE OF LINE 12'
        DFHMDF POS=80,LENGTH=1,COLOR=BLUE,INITIAL=' '
        DFHMDF POS=160,LENGTH=1,COLOR=BLUE,INITIAL=' '
        DFHMDF POS=240,LENGTH=1,COLOR=BLUE,INITIAL=' '
        DFHMDF POS=320,LENGTH=1,COLOR=BLUE,INITIAL=' '
        DFHMDF POS=400,LENGTH=1,COLOR=BLUE,INITIAL=' '
        DFHMDF POS=480,LENGTH=1,COLOR=BLUE,INITIAL=' '
        DFHMDF POS=560,LENGTH=1,COLOR=BLUE,INITIAL=' '
        DFHMDF POS=640,LENGTH=1,COLOR=BLUE,INITIAL=' '
        DFHMDF POS=720,LENGTH=1,COLOR=BLUE,INITIAL=' '
*
BMAP405 DFHMDI SIZE=(3,80),CTRL=(FRSET,FREEKB),COLUMN=1,LINE=22,      *
        MAPATTS=(COLOR,HILIGHT)
FLD00003 DFHMDF POS=80,LENGTH=78,COLOR=BLUE
        DFHMDF POS=160,LENGTH=41,COLOR=BLUE,                          *
        INITIAL='PF1=HELP          PF3=EXIT          PF12=RETURN'
        DFHMDF POS=208,LENGTH=30,COLOR=BLUE,                          *
        INITIAL='ENTER=CONTINUE    CLEAR=EXIT'
BMSSET40 DFHMSD TYPE=FINAL
        END

```



---

## Chapter 24. The Transaction Affinities Utility

You may find it useful to use the Transaction Affinities Utility in the following circumstances to determine whether any transactions in your applications use programming techniques that cause inter-transaction affinity:

- In a CICSplex System Manager/ESA (CICSplex SM) environment for work load balancing
- In a dynamic transaction routing environment for work load balancing
- With user application programs
- If you are planning to implement asynchronous processing using CICS function shipping or transaction isolation.

For detailed information about defining the resources needed by the utility, and about its individual components, see the *CICS Transaction Affinities Utility Guide*.

**Note:** The Transaction Affinity Utility element detects transaction affinities in CICS Transaction Server for OS/390 only. If you want to detect affinities in earlier releases of CICS, for example, CICS/ESA 4.1, you need to use the IBM CICS Transaction Affinities Utility MVS/ESA, program number 5696-582. Contact your IBM representative for more information about this product.



---

## Chapter 25. Offsite Automatic Reply program (DFH\$OFAR)

The offsite automatic reply program (DFH\$OFAR) is a NETVIEW exec that assists in the disaster recovery of a CICSplex when data sets have been used in RLS mode and OFFSITE=YES has been specified as a system initialization parameter.

This utility is needed because the RLS record locks, which preserve data integrity, are not available at the remote site.

---

### Overview of DFH\$OFAR

DFH\$OFAR causes each CICS region to issue message DFHFC0574 to indicate that RLS offsite recovery is being performed, followed by a WTOR message, DFHFC0575, when it has completed recovery of all RLS datasets which that CICS had updated.

The operator is required to wait until every CICS in the CICSplex has issued the message, and only then reply to the DFHFC0575 messages.

This mechanism protects the RLS data sets from being accessed by new work until all the recovery work in the CICSplex has been completed.

A unique control file should exist before DFH\$OFAR is run, which should be accessible from any participating MVS image within the sysplex. The control file of DFH\$OFAR should contain a record for each participating CICS region.

Each participating MVS image in the Sysplex should have NetView configured so that when any CICS region issues messages DFHFC0574 or DFHFC0575, DFH\$OFAR is called.

DFH\$OFAR extracts the relevant input parameters from the message held in the global variables 'token(1/2/..)'. These parameters are the message id, CICS id (APPLID), and the message reply number.

If the message id is DFHFC0574 then DFH\$OFAR updates all entries that are not 'message issued' state to 'message waiting'. Otherwise the existing state is preserved.

If the message id is DFHFC0575 then DFH\$OFAR updates the record for the CICS entry, denoted by the input CICS id, to 'message issued'. If this is not in the control file, it is ignored. All other entries that are not in 'message issued' state are set to 'message waiting'. Otherwise the existing state is preserved.

When all entries in the control file are in 'message issued' state, DFH\$OFAR generates an automatic reply to each DFHFC0575 message issued.

---

### Control file definition

The control file (SYS1.NETVIEW.DFH\$OFAR.CONTROL) should have a data set organization of PS, record format of FB, and record length of 80.

This control file should be accessible from any MVS image that runs a participating CICS.

This control file should be initialized to a list of CICS APPLIDs that are taking part. See the 'Control file typical settings' example on page 208.

---

## Netview configuration

Update the SYS1.PARMLIB member MPFLSTxx ( where xx is the current suffix in use) to include the line:

```
DFHFC057*,AUTO(YES)
```

This causes MVS to invoke Netview whenever a message is issued that is prefixed with DFHFC057.

Add the following entry to DS1PARM, the Netview message table:

```
IF MSGID = 'DFHFC057'. & TEXT = MSG  
THEN EXEC(CMD('DFH$OFAR' MSG) ROUTE(ONE *));
```

This causes Netview to invoke DFH\$OFAR whenever a message is issued that is prefixed with DFHFC057, passing the message text as input parameters.

---

## Control file typical settings

The following are typical settings of the control data set that have been included to help illustrate the usage of DFH\$OFAR:

1. After initial setting by user
  - CICS0001
  - CICS0002
  - CICS0003
2. After CICS0003 has issued message DFHFC0574 (following 1 or 5)
  - CICS0001 MSGWAITING
  - CICS0002 MSGWAITING
  - CICS0003 MSGWAITING
3. After CICS0003 has issued message DFHFC0575
  - CICS0001 MSGWAITING
  - CICS0002 MSGWAITING
  - CICS0003 MSGISSUED 76

Note that '76' represents a typical message reply number
4. After all except CICS0002 have issued message DFHFC0575
  - CICS0001 MSGISSUED 79
  - CICS0002 MSGWAITING
  - CICS0003 MSGISSUED 76
5. After all CICS regions have issued message DFHFC0575
  - CICS0001 MSGREPLIED
  - CICS0002 MSGREPLIED
  - CICS0003 MSGREPLIED

---

## Program failures

All exceptions are returned to the caller with the return code set to one of the following:

- rc = 11 Control data set allocation exceeded 100 attempts
- rc = 12 Control data set reading failed
- rc = 13 Control data set writing failed
- rc = 14 Message reply failure

---

## Chapter 26. 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 written to the CICS local catalog data set. Deleting and adding a subpool record resets the tuning information for that subpool. Deleting and adding a subpool record resets the tuning information for that subpool.

---

### Messages

Messages, including errors, are written to SYSPRINT. DFHSM0300 DFHSMUTL REPORT shows:

1. `ADD SUBPOOL=xxxxxxx PROCESSED SUCCESSFULLY (ADD SUBPOOL=xxxxxxx has been processed successfully.)`
2. `DEL SUBPOOL=xxxxxxx PROCESSED SUCCESSFULLY (DEL SUBPOOL=xxxxxxx has been processed successfully.)`

3. FOUND DFHLCD RECORD SMSUBPOL=xxxxxxx (Subpool record found by the LST command.)
4. ERROR OPENING DFHLCD  
An error has occurred opening the local catalog data set. The program is terminated.
5. UNRECOGNISED VERB xxx IN INPUT (Only ADD, DEL and LST are allowed. The statement is ignored.)
6. UNRECOGNISED OPERAND xxxxxxxx IN INPUT (Only ADD SUBPOOL=xxxxxxx or DEL SUBPOOL=xxxxxxx are allowed. The statement is ignored.)
- 7.

ERROR PROCESSING 'ADD SUBPOOL=xxxxxxx'  
 .R15 = X'yy'.  
 RPL FEEDBACK CODE = X'zz'.  
 SEE DFSMS/MVS<sup>®</sup> 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*. The program is terminated.)

8.

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*. The program is terminated.)

9. 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=CICS320.SDFHLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//DFHLCD DD DSN=CICS320.applid.DFHLCD,DISP=OLD 1
//SYSIN DD *
ADD SUBPOOL=name-of-subpool-to-add 2
DEL SUBPOOL=name-of-subpool-to-remove 3
LST
/*
//
```

**Note:** 1. Change CICS320.applid.DFHLCD 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.



---

## Part 3. Appendixes



---

# Index

## A

- abends
  - cause of immediate shutdown 12
  - CICS system 12
  - during immediate shutdown 14
- ADD command, DFHCSDUP utility program 148
- AID (automatic initiate descriptor)
  - CICS system 13
- ALTER command, DFHCSDUP utility program 149
  - generic naming in 151
- APPEND command, DFHCSDUP utility program 151
  - examples 152
- autoinstall for terminals
  - warm start 7
- automated operations 29
- automatic start 17
- auxiliary trace
  - data sets 94
  - using DFHTU530 to print 93
  - using IPCS to print from GTF 99

## B

- BMS (basic mapping support)
  - warm start 7

## C

- C= keyword, DFHJUP OPTION statement 63
- cataloged procedures
  - starting CICS as a batch job 18
  - starting CICS as a started task 19
- CAVM (CICS availability manager)
  - signoff abnormal at system 14
  - signoff abnormal at system termination 15
- CEMT, master terminal transaction 11, 108
- CEMT transaction
  - PERFORM SHUTDOWN 12, 35
  - PERFORM SHUTDOWN IMMEDIATE 12
- CICS startup
  - overriding startup for START=AUTO 223
- CICS-supplied transactions
  - to control CICS and its resources 11
- CICS system definition file (CSD)
  - offline utility program, DFHCSDUP 139
- CICS system quiesce processing
  - automatic initiate descriptor (AID) 13
  - CLSDST requests for VTAM terminals 13
  - file control 13

- CICS system quiesce processing
  - (continued)
  - first stage 13
  - first stage PLT programs 13
  - interregion communication 13
  - operator notification 13
  - QUIESCE\_DOMAIN calls 13
  - second stage 13
  - second stage PLT programs 13
  - signoff abnormal from CAVM 14
  - subsystem interface 13
  - terminal control 13
  - termination task priority change 13
  - transaction list table (XLT) 13
  - warm keypointing 13
  - warm-start-possible indicator 13
- CICS system shutdown
  - immediate 12, 14
  - normal 12
  - processing 12
  - transaction routing 14
  - types 12
  - uncontrolled 12, 15
  - XRF regions 36
  - XRF systems 12
- CICS system termination
  - first stage processing 14
  - resource managers 14
  - second stage processing 15
  - signoff abnormal from CAVM 15
  - subsystem interface 15
  - terminal control 14
  - termination statistics 14
- CICSDATA, changing to new IPCS verb names 110
- CLIST, TSO command list 29, 195
- cold start
  - DL/I 5
  - file control 5
  - MODEL definitions 5
  - process 5
  - PROFILE definitions 5
  - program control 5
  - PROGRAM definitions 5
  - resource definition 5
  - task control 5
  - terminal control 5
  - TERMINAL definitions 5
  - TRANSACTION definitions 5
  - TYPETERM definitions 5
- compiled modules, scanning 177
- COND= keyword, DFHJUP OPTION statement 63
- console device
  - entering transactions 29
- console devices
  - using TSO command lists 29
- console message-formatting 31
- console support
  - communicating with CICS 29

## consoles

- console messages for CICS
  - startup 20
  - entering system initialization parameters 20
- CONTROL statement, DFHJUP utility program 60
- control statements for DFHDU530 103
- COPY command, DFHCSDUP utility program 153
  - examples 155
  - FROMCSD option 154
  - generic naming in 154
  - MERGE option 154
  - REPLACE option 155
- COPY keyword, DFHJUP OPTION statement 62
- CSA (common system area)
  - warm start 6
- CSD (CICS system definition) file
  - cold start 5
- CSD (CICS system definition file)
  - offline utility program, DFHCSDUP 139

## D

- D= keyword, DFHJUP CONTROL statement 61
- D= keyword, DFHJUP OPTION statement 64
- DDNAME= keyword, DFHJUP CONTROL statement 61
- DDNAME= keyword, DFHJUP OPTION statement 64
- DDNOUT= keyword, DFHJUP CONTROL statement 61
- DEFINE command, DFHCSDUP utility program 155
  - examples 156
- DELETE command, DFHCSDUP utility program 157
  - examples 159
- DFH\$MOLS sample print program 119
  - control statements 128, 129
    - BREAK 129
    - comments 129
    - DATE 131
    - IGNORE 131
    - OPTION 132
    - PRINT 133
    - SELECT 133
    - SORT 134
  - monitoring sample utility 124
  - overview 124
  - sample job stream for 125
- DFH\$STED, utility to stagger the end-of-day time 197
- DFH\$STER
  - recovery statistics program 91
  - statistics sample program 92

- DFHAUXT auxiliary trace data set 94
  - DFHBMSUP utility program
    - introduction 229
  - DFHBUXT auxiliary trace data set 94
  - DFHCESD, shutdown assist program
    - default actions 217
    - introduction 217
    - sample programs 218
  - DFHCSDUP system definition utility program 139, 144
    - ADD command 148
    - ALTER command 149
    - APPEND command 151
    - command processing
      - considerations 148
    - COPY command 153
    - DEFINE command 155
    - DELETE command 157
    - EXTRACT command 160
    - INITIALIZE command 161
    - invocation from a user program 144
    - invoking as a batch program 141
    - LIST command 162
    - MIGRATE command 164
    - PROCESS command 168
    - processing system definition file 139
    - REMOVE command 169
    - running under TSO 144
    - SCAN command 170
    - SERVICE command 172
    - UPGRADE command 173
    - VERIFY command 174
  - DFHCSVC, the CICS type 3 SVC 23
  - DFHDU530 transaction dump utility program
    - INDEX DD statement for dump
      - summary 103
    - job control statements for 106
    - processing transaction dump data sets 103
    - SYSIN control statements 103
  - DFHIPCSP, IPCS exit control table member for CICS 109
  - DFHKE1799 message 14, 15, 35, 36
  - DFHLSCU 41
    - DFHLSCU SYSIN statements
      - AKPFREQ 45
      - INTERVAL 45
      - JNLTYPE 44
      - LOGSNUM 45
      - TRANS DUR 45
    - JCL 43
  - DFHMNDUP dictionary utility program 119
  - DFHMSCAN utility program, to identify macro-level programs
    - detailed report 193
    - how the program works 192
    - JCL 191
    - limitations of 194
    - overview 191
    - summary report 192
  - DFHPD530, SDUMP formatting exit routine for IPCS 110
    - sample jobs for formatting CICS SDUMPs 116
  - DFHPDX, SDUMP formatting exit routine of IPCS 110
  - DFHRMUTL, recovery manager utility program
    - introduction 223
  - DFHSMUTL local catalog utility program. 237
  - DFHSNMIG utility program, to migrate from SNT to RACF database 195
    - sample job stream to execute CLIST 196
    - sample job stream to run utility program 196
  - DFHSTUP statistics utility program
    - control parameters 83
    - job stream for 78
    - printing CICS statistics 77
  - DFHTU530 trace utility program
    - printing auxiliary trace data 93
    - selection parameters
      - ABBREV 95
      - ALL 95
      - ENTRY\_NUM 95
      - EXCEPTION 95
      - FULL 95
      - KE\_NUM 96
      - PAGESIZE 94, 96
      - SHORT 95
      - TASKID 96
      - TERMID 96
      - TIMERG 96
      - TRANID 96
      - TYPETR 97
    - diagnostic run, of CICS 225
    - dump exit routines 110, 111
      - sample jobs for formatting CICS SDUMPs 116
    - dumps
      - CICS transaction dumps 103
      - DFHDU530 utility program, to process CICS transaction dumps 103
    - DFHIPCSP, exit control table member for CICS 109
    - DFHPD530, SDUMP formatting exit routine of IPCS 110
    - dump summary 115
      - example 115
    - error index 115
    - exit parameters for CICS 111, 112
    - exit routines for IPCS 110, 111
    - formatting with IPCS 111
    - IPCS exit control table 109
    - IPCS verb names for CICS 110
    - sample formatting jobs using IPCS 116
    - SDUMP formatting exit routines of IPCS 110
    - SDUMP options needed for the CICS dump exit 109
    - VERBEXIT command of IPCS 111
    - via MVS SDUMP macro 103
      - processing using IPCS 108
    - What to do before formatting with IPCS 109
  - dynamic transaction backout
    - failure during immediate shutdown 14
- ## E
- E= keyword, DFHJUP OPTION statement 63
  - editing messages 199
  - emergency restart
    - process 8
    - resynchronization of VTAM messages 8
  - EXEC CICS PERFORM SHUTDOWN command 12, 35
  - EXEC CICS PERFORM SHUTDOWN IMMEDIATE command 12
  - EXITR= keyword, DFHJUP OPTION statement 63
  - extended storage cushion
    - warm start 6
  - EXTRACT command, DFHCSDUP utility program 160
    - examples 161
    - OBJECTS option 160
    - USERPROGRAM option 160
- ## F
- FCT (file control table)
    - warm start 6
  - file control
    - file states on warm start 6
    - warm start 6
  - filter input file
    - definition 183
  - FLDLEN= keyword, DFHJUP OPTION statement 63
  - FLDTYP= keyword, DFHJUP OPTION statement 62
- ## G
- generalized trace facility (GTF)
    - formatting trace records 100
    - GTFTRACE subcommand of IPCS 100
    - printing CICS traces using IPCS 99
    - sample job to print GTF trace entries 101
  - generic naming in the ALTER command 151
  - generic naming in the COPY command 154
  - global catalog
    - file control table (FCT) warm start 6
    - monitoring options 7
    - statistics options 7
    - warm keypointing during CICS system 13
    - warm-start-possible indicator 13
    - warm start resource definition 5
- ## H
- H= keyword, DFHJUP CONTROL statement 60

## I

ICE (interval control element)  
warm start 7

INITIALIZE command, DFHCSDUP  
utility program 161

interactive problem control system (IPCS)  
DFHIPCSP, IPCS exit control table  
member for CICS 109  
dump summary 115  
example 115  
error index 115  
IPCS verb names for CICS 110  
printing CICS traces from GTF 99  
processing CICS SDUMPs 108  
sample formatting jobs 116  
SDUMP options needed for the CICS  
dump exit 109

interregion communication (IRC) 26

intrapartition data set  
warm start 6

IRC (interregion communication) 26  
CICS system 13

## J

JCL to submit CICS commands 30

job control language (JCL)  
for CICS as a batch job 18  
for CICS as a started task 19

journal utility program (DFHJUP)  
control statements  
COMMENTS 65  
CONTROL 60  
END 64  
OPTION 61  
input and output 57

## K

K= keyword, DFHJUP CONTROL  
statement 60

kernel domain  
CICS system termination  
processing 14, 35, 36

keywords  
C=, of DFHJUP OPTION  
statement 63  
COND=, of DFHJUP OPTION  
statement 63  
COPY, of DFHJUP OPTION  
statement 62  
D=, of DFHJUP CONTROL  
statement 61  
D=, of DFHJUP OPTION  
statement 64  
DDNAME=, of DFHJUP CONTROL  
statement 61  
DDNAME=, of DFHJUP OPTION  
statement 64  
DDNOUT=, of DFHJUP CONTROL  
statement 61  
E=, of DFHJUP OPTION  
statement 63  
EXITR=, of DFHJUP OPTION  
statement 63

## keywords (continued)

FLDLEN=, of DFHJUP OPTION  
statement 63

FLDTYP=, of DFHJUP OPTION  
statement 62

H=, of DFHJUP CONTROL  
statement 60

K=, of DFHJUP CONTROL  
statement 60

L=, of DFHJUP OPTION  
statement 63

NEWDCB, of DFHJUP OPTION  
statement 64

O=, of DFHJUP CONTROL  
statement 61

O=, of DFHJUP OPTION  
statement 62

OFFSET=, of DFHJUP OPTION  
statement 62

P=, of DFHJUP OPTION  
statement 64

PRINT, of DFHJUP OPTION  
statement 62

PRTSYS=, of DFHJUP OPTION  
statement 64

SKIP=, of DFHJUP CONTROL  
statement 60

STOPAFT=, of DFHJUP CONTROL  
statement 60

T=, DFHJUP OPTION statement 62

V=, of DFHJUP OPTION  
statement 62

VALUE=, of DFHJUP OPTION  
statement 62

## L

L= keyword, DFHJUP OPTION  
statement 63

LIST command, DFHCSDUP utility  
program 162  
examples 163  
OBJECTS option 163

load module scanner  
activating trace 184  
detailed report 179  
summary report 178

load module scanner, utility  
program 177

local catalog utility program,  
DFHSMUTL 237

log stream and coupling facility sizing,  
DFHLSCU 41

## M

machine check 12

macro-level programs, identifying 191

mapset definitions  
warm start 7

master terminal transaction, CEMT 11

maximum task values  
warm start 6

message editing utility 199  
defining the utility data set  
index 200

message editing utility 199 (continued)  
edit message panel 208  
help panels 215  
installing 199  
language selection panel 203  
main panel 205  
message edit panel 212  
message edit selection panel 207  
PTF update panel 210  
requirements 199  
set defaults panels 201  
using 200  
applying PTFS 210  
applying PTFS, rules 211  
applying PTFS, sample  
output 211  
copying message data set  
members 206  
editing messages 208  
editing messages, rules to be  
observed 212  
generating message load  
modules 209  
getting help 215  
link editing changed message  
source members 209  
performing actions on message  
data sets 205  
selecting languages for  
translation 203  
selecting message sets to be  
edited 207  
sorting lists of message set  
members 209  
specifying default values 201  
starting 201

messages  
console messages for CICS  
startup 20  
DFHKE1799 14, 15, 35, 36  
replying to messages 32  
replying to messages from  
transactions 32  
suppressing 32  
suppressing and rerouting 33  
VTAM 8

MIGRATE command, DFHCSDUP utility  
program 164  
DCT migration 165  
FCT migration 165  
RCT migration 166  
TABLE option 168  
TCT migration 166  
TOGROU option 168  
TST migration 167  
TYPESGROUP option 168  
TYPETERM creation 167

migration utility program, SNT to  
RACF 195

MODEL definitions  
cold start 5

MODIFY command 29

monitoring  
dictionary utility program,  
DFHMNDUP 119  
sample print program,  
DFH\$MOLS 119

monitoring (*continued*)  
warm start 7  
MRO (multiregion operation) 26  
multiregion operation (MRO) 26  
MVS START command, to start CICS 19

## N

NEGO keyword, DFHJUP OPTION  
statement 62  
NEWDCB, DFHJUP OPTION  
statement 64

## O

O= keyword, DFHJUP CONTROL  
statement 61  
O= keyword, DFHJUP OPTION  
statement 62  
OBJECTS option  
LIST command (DFHCSDUP) 163  
OFFSET= keyword, DFHJUP OPTION  
statement 62  
Offsite automatic reply program 235  
operating system failure 12  
operations, automated 29  
operator communication for initialization  
parameters 21  
overriding system initialization  
parameters  
from the console 20

## P

P= keyword, DFHJUP OPTION  
statement 64  
PLT (program list table)  
first stage shutdown programs 13  
second stage shutdown programs 13  
power failure 12  
PRINT keyword, DFHJUP OPTION  
statement 62  
PROCESS command, DFHCSDUP utility  
program 168  
PROFILE definitions  
cold start 5  
warm start 6  
program check  
cause of immediate shutdown 12  
PROGRAM definitions  
cold start 5  
warm start 7  
program library  
cold start resource definition 5  
warm start resource definition 5  
programs  
PLT first stage shutdown 13  
PLT second stage shutdown 13  
PRTSYS= keyword, DFHJUP OPTION  
statement 64  
PTFs, DFHMEU update log (sample  
output) 211

## R

RACF (resource access control facility)  
migrating entries from the CICS  
SNT 195  
recovery manager domain 91  
recovery manager utility program,  
DFHRMUTL  
introduction 223  
recovery messages  
Recovery Manager domain 91  
recovery statistics program,  
DFH\$STER 91  
region exit time interval value  
warm start 6  
REMOVE command, DFHCSDUP utility  
program 169  
examples 170  
REPLY command, responding to console  
messages 32  
resend slot  
re-presentation of VTAM messages 8  
resource access control facility (RACF)  
migrating entries from the CICS  
SNT 195  
resource definition  
cold start 5  
warm start 5  
runaway timer interval value  
warm start 6

## S

SCAN command, DFHCSDUP utility  
program 170  
SDTRAN, system initialization  
parameter 217  
SERVICE command, DFHCSDUP utility  
program 172  
FROMCSD operand 173  
LEVEL operand 173  
shutdown assist program, DFHCESD  
default actions 217  
introduction 217  
sample programs 218  
SKIP= keyword, DFHJUP CONTROL  
statement 60  
SNT to RACF migration utility  
program 195  
staggering the end-of-day time 197  
START, system initialization  
parameter 223  
START=AUTO 17  
START=COLD 18  
START=INITIAL 18  
START=STANDBY 18  
START command, MVS 19  
started task, CICS as a 19  
starting CICS regions 3, 17  
as a batch job 18  
as a started task 19  
MVS START command 19  
START=AUTO 17, 18  
START=INITIAL 18  
START=STANDBY, for an XRF  
alternate CICS 18

statistics  
termination 14  
warm start 7  
statistics, staggering the end-of-day  
time 197  
statistics sample program,  
DFH\$STER 92  
statistics utility program, DFHSTUP 77  
STOPAFT= keyword, DFHJUP  
CONTROL statement 60  
storage cushion  
warm start 6  
STORECLOCK value warm start 7  
subsystem interface  
termination 13, 15  
SVC  
specifying DFHC SVC in the startup  
job 23  
system initialization  
for an alternate CICS (XRF=YES)  
START=STANDBY 18  
START=AUTO 17, 18  
START=INITIAL 18  
system initialization parameters  
entering at the console 20  
from operator's console 21  
SDTRAN 217  
START 223  
system startup 3, 17  
system task 19

## T

T= keyword, DFHJUP OPTION  
statement 62  
TCT (terminal control table)  
warm start 7  
temporary storage 7  
warm start 7  
terminal control  
CICS system processing 13  
termination processing 14  
TERMINAL definitions  
cold start 5  
termination task  
priority change to zero 13  
processing 13  
time sharing option (TSO)  
the DFHCSDUP program 144  
using command lists 29  
trace  
using DFHTU530 to print 93  
using IPCS to print from GTF 99  
utility programs 93  
trace utility program, DFHTU530 93  
transactionabend  
during immediate shutdown 14  
Transaction Affinities Utility 233  
TRANSACTION definitions  
cold start 5  
warm start 6  
transaction list table (XLT) 13  
transient data  
intrapartition warm start 6  
translating messages 199  
TSO (time sharing option)  
the DFHCSDUP program 144

TSO (time sharing option) (*continued*)  
  using command lists 29  
tuning the system 11  
TYPETERM definitions  
  cold start 5

## U

unit of recovery descriptor (URD) 7  
UPGRADE command, DFHCSDUP utility  
  program 173  
  REPLACE operand 174  
  USING operand 174  
URD (unit of recovery descriptor) 7  
utility program, to scan for API  
  commands  
  overview 177  
utility programs 12  
utility programs, offline  
  assisting in disaster recovery of a  
  CICSplex, DFH\$OFAR 235  
  batch-enabling sample programs for  
  RLS access-mode data sets 187  
  change text or language of CICS  
  messages 199  
  default shutdown assist program,  
  DFHCESD 217  
  detecting transaction affinities 233  
  identifying macro-level programs  
  (DFHMSCAN) 191  
  initializing CICS system definition file,  
  DFHCSDUP 139  
  running under TSO 144  
  load module scanner, scanning for  
  API commands 177  
  log stream and coupling facility  
  sizing, DFHLSCU 41  
  migrating SNT entries to the RACF  
  database (DFHSNMIG) 195  
  preparing statistics reports,  
  DFHSTUP 77  
  preparing trace reports,  
  DFHTU530 93  
  processing CICS monitoring data,  
  DFH\$MOLS 119, 124  
  processing CICS monitoring data,  
  DFHMNDUP 119  
  processing log data, DFHJUP 53  
  processing the global catalog dataset,  
  DFHRMUTL 223  
  processing transaction dump data sets  
  (DFHDU530) 103  
  recreating BMS macro statements,  
  DFHBMSUP 229  
  staggering the end-of-day time  
  (DFH\$STED) 197

## V

V= keyword, DFHJUP OPTION  
  statement 62  
VALUE= keyword, DFHJUP OPTION  
  statement 62  
VERBEXIT subcommand of IPCS 108,  
111  
  exit parameters for CICS 111, 112

VERIFY command, DFHCSDUP utility  
  program 174  
VTAM messages  
  resynchronization after emergency  
  restart 8

## W

warm keypoints  
  warm start resource definition 5  
warm start  
  autoinstalled terminals 7  
  basic mapping support (BMS) 7  
  common system area (CSA) 6  
  file control table (FCT) 6  
  file states 6  
  installed program definitions 7  
  interval control elements 7  
  intrapartition transient data 6  
  logical end of day 7  
  mapset definitions 7  
  monitoring 7  
  partial 5  
  process 5  
  profile definitions 6  
  resource definition 5  
  statistics collecting interval 7  
  statistics collecting status 7  
  STORECLOCK value 7  
  temporary storage 7  
  terminal control table (TCT) 7  
  transaction definitions 6  
  unit of recovery descriptor (URD) 7  
warm-start-possible indicator 13

## X

XLT (transaction list table) 13  
XMEOU, global exit for message  
  handling 33  
XRF (extended recovery facility)  
  shutdown of active 12, 36  
  termination of alternate 12, 36  
  VTAM ACB at startup 8



---

## Sending your comments to IBM

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, to this address:

Information Development Department (MP095)  
IBM United Kingdom Laboratories  
Hursley Park  
WINCHESTER,  
Hampshire  
SO21 2JN  
United Kingdom

- By fax:
  - From outside the U.K., after your international access code use 44-1962-870229
  - From within the U.K., use 01962-870229
- Electronically, use the appropriate network ID:
  - IBM Mail Exchange: GBIBM2Q9 at IBMMAIL
  - IBMLink™ : HURSLEY(IDRCF)
  - Internet: idrcf@hursley.ibm.com

Whichever you use, ensure that you include:

- The publication number and title
- The topic to which your comment applies
- Your name and address/telephone number/fax number/network ID.



Program Number: 5655-147



Printed in the United States of America  
on recycled paper containing 10%  
recovered post-consumer fiber.

SC33-1685-33



Spine information:



CICS TS for OS/390

CICS Operations and Utilities Guide

Release 3