CICS Transaction Server for z/OS



Utilities Reference

Version 5 Release 4

CICS Transaction Server for z/OS



Utilities Reference

Version 5 Release 4

Note
Before using this information and the product it supports, read the information in "Notices" on page 155.

This edition applies to the IBM CICS Transaction Server for z/OS Version 5 Release 4 (product number 5655-Y04) and to all subsequent releases and modifications until otherwise indicated in new editions.

© Copyright IBM Corporation 1989, 2017.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About this PDF. v

Chapter 1. Reading log streams using batch jobs (for example, DFHJUP) . . . 1

SUBSYS=(LOGR,DFHLGCNV,...) keyword 1
Using the SETBRCUR, REPBRCUR and DELBRCUR options for log stream processing . . 8
Using DFHJUP to read log streams 9
DD statements for DFHJUP or your own batch job 10
Utility control statements for DFHJUP 11
CONTROL statement 12
OPTION statement 13
Options. 13
END statement 17
COMMENTS statement 18
DFHJUP return codes 18
Managing the size of log streams 18
System log 18
General logs 18
Log data accessible to DFHJUP 19
Example of log data that is accessible to DFHJUP 19
Diagnostic information in DFHJUP output 20
Examples of using DFHJUP 21
Example 1 21
Example 2 22
Example 3 23
Example 4 23
Examples of the use of the OPTION parameters 24
Version 5 Release 4 format 25
COMPAT41 format 26
OPTION parameter values 28

Chapter 2. Statistics utility program (DFHSTUP) 31

Setting the statistics recording status, end-of-day time and recording interval 31
Support for Katakana devices 33
Job to run the DFHSTUP program. 33
The DFHSTUP extract statistics reporting function 36
The DFH0STXR sample program 36
The DFH0STXD sample program 38
Using the extract statistics reporting function . . 39
Control parameters of the DFHSTUP program. . . 42

Chapter 3. Recovery statistics program (DFH$STER) 51

Recovery Manager domain recovery messages. . . 51
Statistics sample program, DFH$STER 52
Installing DFH$STER 52

Chapter 4. Trace utility print program (DFHTU710) 53

The CICS trace utility program, DFHTU710 . . . 53
The trace selection parameters for DFHTU710 . . . 54
Identifying trace entries from their transaction-attach entries 59
Rules for coding trace selection parameters . . . 60
Using IPCS to print trace records written to GTF . . 61
The GTFTRACE subcommand of IPCS and associated parameters 62
Sample batch job to print CICS GTF trace entries 63
Invoking the CICS formatting routine from TSO 63

Chapter 5. Dump utilities (DFHDU710 and DFHPD710) 65

Selecting output for the transaction dump utility program (DFHDU710) 65
Format of the SYSIN control statements 66
Job control statements for the transaction dump utility program (DFHDU710) 69
Using IPCS to format and analyze CICS dumps: Overview 70
Preparing to use IPCS to format CICS SDUMPs . . 71
The SDUMP options needed to support the CICS dump exit 71
Specifying DFHIPCSP CICS exit control data needed 72
Making the required CICS dump exit routines available 72
Selecting parts of the CICS internal trace table. . . 73
Using CICS-supplied dump exit routines to format CICS SDUMPs 74
The CICS dump exit parameters 74
Exceptions to the scope of the DEF parameter . . 75
Examples of the use of level numbers 76
The CICS710 dump exit component keywords. . 76
The dump summary and error index 77
Sample jobs to process a CICS SDUMP using the CICS dump exit 78

Chapter 6. Monitoring dictionary utility program (DFHMNDUP) 83

Generating a performance dictionary record using DFHMNDUP. 83
Parameters for the DFHMNDUP program 85
Sample job illustrating the use of DFHMNDUP . . 86

Chapter 7. Sample monitoring data print program (DFH$MOLS) 89

Processing CICS monitoring data 90
Sample job stream for DFH$MOLS 91
DFH$MOLS control statements summary 92
Rules for coding DFH$MOLS control statements . . 94
DFH$MOLS control statement descriptions 94
Abend codes and error messages for DFH$MOLS 102
101: INVALID, DUPLICATE, OR MISSING CONTROL CARD INFORMATION 102

© Copyright IBM Corp. 1989, 2017

102: ERROR - INCOMPATIBLE CONTROL CARD OPTIONS REQUESTED 102
103: SORT ENDED WITH NON-ZERO RET CODE; RC='nn'; REPORT IS TERMINATED . . 102
104: NO DATA FOR THIS CICS DATA SECTION; REPORT IS TERMINATED 103
105: UNKNOWN CLASS IN CICS DATA SECTION; REPORT IS TERMINATED 103
106: NO VALID MONITORING RECORDS WERE READ; REPORT IS TERMINATED . . . 103
107: NO MONITORING RECORDS WERE SELECTED FOR PROCESSING; REPORT IS TERMINATED 103
108: X'nnnn' - CONNECTOR NOT DEFINED; REPORT IS TERMINATED 103
109: NO DICTIONARY ESTABLISHED; REPORT IS TERMINATED 104
110: NO DICTIONARY ESTABLISHED FOR NEW APPLID xxxxxxxx; REPORT IS TERMINATED 104
112: There is no message text, but MVS abends U112 without a dump. 104
113: UNABLE TO OPEN DDNAME 'INPUT'; REPORT IS TERMINATED 104
114: UNABLE TO OPEN DDNAME 'xxxxxxxx'; REPORT IS TERMINATED 105
116: NO FILE RESOURCE ENTRIES. 105
117: NO TSQUEUE RESOURCE ENTRIES . . . 105
118: UNABLE TO EXPAND A COMPRESSED RECORD, RC='nn'; REPORT IS TERMINATED . 105
119: UNABLE TO OPEN DDNAME '*xxxxxxxx*'; REPORT IS TERMINATED 105
120: UNEXPECTED CSRCESRV QUERY ERROR, RC='nn'; REPORT IS TERMINATED. . 106
121: NO DPL RESOURCE ENTRIES 106

Chapter 8. Load module scanner (DFHEISUP) 107
Creating a filter input file for the load module scanner 108
Creating a summary report using the load module scanner 111
Creating a detailed report using the load module scanner 114
Activating trace for the load module scanner . . . 117

Chapter 9. Batch-enabling sample programs for RLS access-mode data sets (DFH0BATx) 119
Preparing data sets for batch operations using the DFH0BATx sample programs 120
Installing the DFH0BATx sample programs . . 122
Preparing input for the DFH0BATx sample programs. 122

Chapter 10. Identify macro-level programs utility program (DFHMSCAN) 125
How DFHMSCAN works 125
JCL to run DFHMSCAN. 126
Producing a summary report from DFHMSCAN 126
Producing a detailed report from DFHMSCAN . . 127
Limitations of the DFHMSCAN program 128

Chapter 11. Stagger end-of-day time sample utility program (DFH$STED). . 129
Example values for parameters of DFH$STED . . 130

Chapter 12. Shutdown assist program (DFHCESD) 131
Actions of the default program, DFHCESD . . . 131
The sample shutdown assist programs 132

Chapter 13. Recovery manager utility program (DFHRMUTL) 137
JCL requirements for DFHRMUTL 138
Specifying parameters for DFHRMUTL. 138
Return codes from DFHRMUTL 140
Examples of using DFHRMUTL 141
Setting an initial start without operator intervention 141
Examining the override record 141
Resetting a warm or emergency start 142
Improving the performance of a cold start. . . 142

Chapter 14. BMS macro generation utility program (DFHBMSUP). 145
DD statements for DFHBMSUP 145
Return codes from DFHBMSUP 145
Example of using DFHBMSUP 146
Example of DFHBMSUP output 146

Chapter 15. Offsite Automatic Reply program (DFH$OFAR). 149
DFH$OFAR control file definition 149
NetView configuration 150
DFH$OFAR control file typical settings. 150
Return codes from DFH$OFAR 150

Chapter 16. Local catalog storage program (DFHSMUTL). 153
Job control statements to run DFHSMUTL. . . . 153
Messages from DFHSMUTL 154

Notices 155

Index 161

About this PDF

This PDF is a reference of the utilities that support the operation of CICS. This PDF is a companion to *Administering CICS*. Before CICS TS V5.4, the information in this PDF (and the information in *Administering CICS*) was in the *Operation and Utilities Guide*.

For details of the terms and notation used in this book, see Conventions and terminology used in the CICS documentation in IBM Knowledge Center.

Date of this PDF

This PDF was created on June 6th 2017.

© Copyright IBM Corp. 1989, 2017

Chapter 1. Reading log streams using batch jobs (for example, DFHJUP)

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.

About this task

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.

When reading log stream data, be careful not to discard records prematurely. Records returned to a user application program are considered to have been processed, and the appropriate log stream cursor (the browsed cursor, the deleted cursor, or the LASTRUN cursor) is updated. On subsequent runs, if you use cursor-related processing options, the records you processed previously are not read. Do not assume that records returned from DFHJUP or your own batch job will be seen again on a later run, as this might not be the case. Only discard such records if you are sure that you no longer need them on the log stream.

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 its input or data DD.

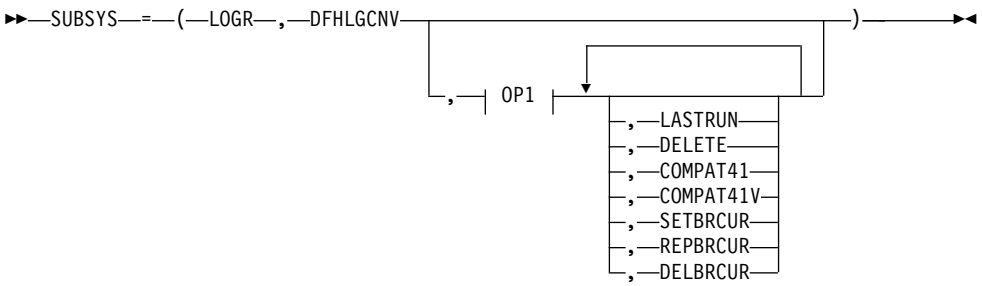
SUBSYS=(LOGR,DFHLGCNV,...) keyword
has the form and syntax described in Figure 1 on page 2:

© Copyright IBM Corp. 1989, 2017

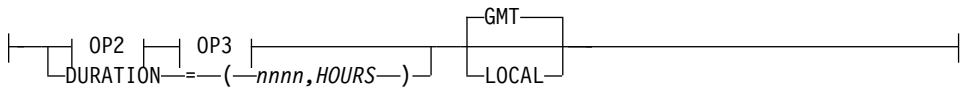
```
//ddname   DD  DSNAME=log_stream_name,
//            SUBSYS=
```

where SUBSYS expands as follows:

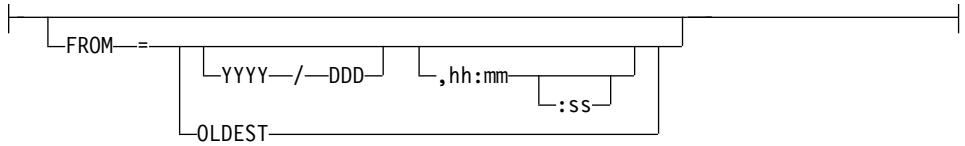
SUBSYS



OP1:



OP2:



OP3:

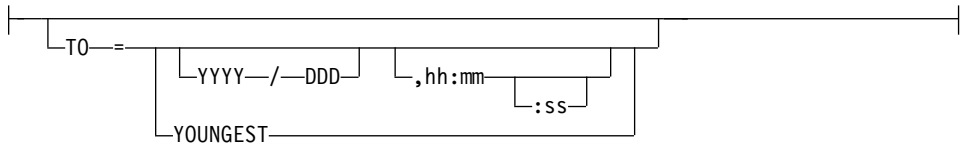


Figure 1. Log stream SUBSYS data set specification

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 the 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.

Note: If multiple options are required, quotation marks are required around each set of options, and the set of options should be separated by a comma. Quotation marks are not required when only one option is required. For example, the coding `SUBSYS=(LOGR,DFHLGCNV,TO=(,12:00),COMPAT41)` uses only one SUBSYS-options1 option and one SUBSYS-options2 option, so it does not require quotation marks. However, the coding `SUBSYS=(LOGR,DFHLGCNV,'TO=(,12:00),GMT','COMPAT41,SETBRCUR')` uses two SUBSYS-options1 options and two SUBSYS-options2 options. Quotation marks are used around each set of options, and the sets of options are separated by a comma.

'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 steam 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.

GMT|LOCAL
specifies whether the time is local time or GMT time. GMT is the default.

SUBSYS-options2
specifies exit routine unique options. Table 1 on page 7 shows the combinations of SUBSYS-options2 options that are supported.

Note: If multiple options are required, quotation marks are required around each set of options, and the set of options should be separated by a comma. Quotation marks are not required when only one option is required. For example, the coding
`SUBSYS=(LOGR,DFHLGCNV,TO=(,12:00),COMPAT41)` uses only one SUBSYS-options1 option and one SUBSYS-options2 option, so it does not require quotation marks. However, the coding
`SUBSYS=(LOGR,DFHLGCNV,'TO=(,12:00),GMT','COMPAT41,SETBRCUR')` uses two SUBSYS-options1 options and two SUBSYS-options2 options. Quotation marks are used around each set of options, and the sets of options are separated by a comma.

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 FROM, TO and DURATION 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

Use the SETBRCUR, REPBRCUR and DELBRCUR options for log stream processing. For more information about how to do this see “Using the SETBRCUR, REPBRCUR and DELBRCUR options for log stream processing” on page 8.

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 before 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 22.

COMPAT41 or COMPAT41V
If you are running applications against the log stream that require the records to be presented, as far as possible, in the format introduced at CICS Transaction Server for OS/390®, Version 1 Release 1, specify the COMPAT41 option. For COBOL applications, specify the COMPAT41V option. This ensures that records are presented in the correct variable blocked format.

Alternatively, you can create a new data set with the correct DCB information and the records, as far as possible, in the format used by CICS/ESA 4.1. To do this, run DFHJUP specifying the COMPAT41 option (and not the COMPAT41V option), and also specifying the NEWDCB option on the OPTION COPY command. (See "Utility control statements for DFHJUP" on page 11 for more information on creating a new data set using NEWDCB.) You can then run batch applications that require the records in COMPAT41 format against this new data set. Note that once you have created a new data set, the batch applications do not need the COMPAT41 or COMPAT41V option to be specified at runtime.

SETBRCUR
indicates that the starting point of the records to be read from the log stream is the position where a deleted cursor was last set (by a batch program that used the DELBRCUR option). Records are read from the deleted cursor to the youngest block in the log stream, or to another position that you have specified (for example, using the TO keyword). When the required records have been read, a browsed cursor is positioned at the last record read.

To ensure that you are not reading the same records twice, after you run each job that uses the SETBRCUR option, run a job using the DELBRCUR option to delete the browsed records and reposition the deleted cursor at the last record deleted. If you do not reposition the deleted cursor, any subsequent job using SETBRCUR starts to read from the same record as the first job.

If you do want to read the same set of records twice, run a job using the SETBRCUR option followed by a job using the REPBRCUR option. This combination of jobs produces two copies of exactly the same records, so you can produce two copies of the same records in different formats, for example.

Do not use the SETBRCUR option with the DELETE or LASTRUN options.

REPBRCUR
indicates that records are to be read from the log stream from the position where a deleted cursor was last set (by a batch program that used the DELBRCUR option), to the position where a browsed cursor was last set (by a batch program that used the SETBRCUR option). It repeats exactly the same data that was processed last time you ran a batch program using the SETBRCUR option, provided that the deleted

cursor has not been moved since that read. The deleted cursor and browsed cursor are not changed by running a batch program using the REPBRCUR option.

The REPBRCUR option can be used to produce a second copy of the same records, perhaps in a different format to that produced by the original processing. To do this, run a job using the SETBRCUR option followed by a job using the REPBRCUR option.

Do not use the REPBRCUR option with the DELETE or LASTRUN options.

DELBRCUR
indicates that log stream records are to be deleted from the log stream as far as the position where a browse cursor was last set (by a batch program that used the SETBRCUR option). When the required records have been deleted, a deleted cursor is placed at the position where the browse cursor was set.

To ensure that you are not reading the same records twice, after you run each job that uses the SETBRCUR option, run a job using the DELBRCUR option to reposition the deleted cursor. If you do not reposition the deleted cursor, any subsequent job using SETBRCUR starts to read from the same record as the first job. If you do want to read the same set of records twice, run a job using the REPBRCUR option, before you run a job using the DELBRCUR option to reposition the deleted cursor.

Do not use the DELBRCUR option with any other options.

Table 1 shows the combinations of SUBSYS options that are supported. In each row of the table, you can use any combination of the options shown, including all or none of them. The exception to this rule is the time-related options TO, FROM and DURATION. You can use TO with FROM, and FROM with DURATION, but not TO with DURATION. You can use any of the time-related options on their own.

SETBRCUR, REPBRCUR and DELBRCUR are the recommended options for log stream offline processing, for example for archive purposes. For more information, see "Using the SETBRCUR, REPBRCUR and DELBRCUR options for log stream processing" on page 8.

Table 1. SUBSYS - combinations of options

Starting point	Compatibility	Deletion	Time
	COMPAT41		TO, FROM or DURATION
	COMPAT41V		TO, FROM or DURATION
LASTRUN	COMPAT41		TO or DURATION
LASTRUN	COMPAT41V		TO or DURATION
SETBRCUR	COMPAT41		TO or DURATION
SETBRCUR	COMPAT41V		TO or DURATION
REPBRCUR	COMPAT41		
REPBRCUR	COMPAT41V		
		DELETE	TO or DURATION
		DELBRCUR	

Using the SETBRCUR, REPBRCUR and DELBRCUR options for log stream processing

SETBRCUR, REPBRCUR and DELBRCUR are the recommended options for log stream offline processing.

About this task

Using these options ensures that the beginning and end of your read are precisely defined, and the reads are repeatable. For example, using the REPBRCUR option enables you to archive the same records in two different formats. These options also provide enhanced support for automatic logstream processing, as you do not need to include date and time information in your batch program. Note that log streams to be processed in this way should be defined as AUTODELETE(NO) RETPD(0), to avoid unexpected loss of data.

It is recommended that you should not run more than one job per logstream using a SETBRCUR, REPBRCUR or DELBRCUR option at any one time in your sysplex. If you run more than one such job simultaneously, they could update the same cursor, and then your reads would not be repeatable. You should also not run a job using LASTRUN at the same time as a job using SETBRCUR, REPBRCUR or DELBRCUR. You can run jobs that use only non-cursor related options at the same time as a job using one of the cursor-related options (SETBRCUR, REPBRCUR, DELBRCUR and LASTRUN), as the other log stream processing options do not affect the position of the cursors.

A suggested sequence for using the SETBRCUR, REPBRCUR and DELBRCUR options for log stream processing is as follows:

Procedure

1. Run a job using the SETBRCUR option, to read a set of records and then set the browsed cursor at the last record read.
2. If the job fails, you can repeat the job using the SETBRCUR option.
3. If you want to re-read the same set of records, for example to produce the same records in different formats, run a job using the REPBRCUR option. This option does not change the position of any cursors.
4. If the job fails or you want to re-read the same set of records again, you can repeat the job using the REPBRCUR option.
5. If you want to repeat the whole reading process thus far for any reason, you can now start again at Step 1. If you restart the process, any new data that has been placed on the log stream can be picked up.
6. If the reading process has been successful and you are certain that you no longer need the records that you have read, run a job using the DELBRCUR option, to delete the records that you have processed and then set the deleted cursor at the end of the records you have just deleted.
7. If the job fails, you can repeat the job using the DELBRCUR option.
8. You have now processed a set of records and deleted them from the log stream.

Example

The following flowchart illustrates this sequence of actions:

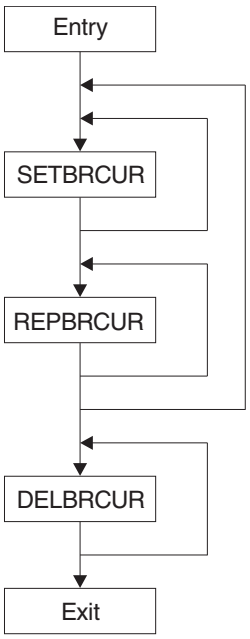


Figure 2. Log stream processing using SETBRCUR, REPBRCUR and DELBRCUR

For an example of how you can use SETBRCUR and REPBRCUR to make two copies of the same log stream partition, then delete the log stream partition using DELBRCUR, see "Example 4" on page 23.

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 can use multiple input log streams or SMF data sets, and format the output for multiple output data sets.

About this task

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 21 gives some sample jobs that illustrate the use of DFHJUP.

DD statements for DFHJUP or your own batch job

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 default ddname is SYSUT1.

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.

DASD-only log streams have the added restriction that they cannot be shared between MVS images; they can only be accessed from one MVS image at a time (although multiple connections are permitted from that MVS image). This means that if a DASD-only log stream is being accessed from a particular MVS image in the sysplex (for example, by a CICS region), DFHJUP must also run in that MVS image. If the DASD-only log stream is not currently being accessed, DFHJUP can run in any of the MVS images in the sysplex.

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 1 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 DSNAME=CICSLOG,DISP=(OLD,KEEP),
//             DCB=RECFM=VB
```

The second example shows the use of the BLKSIZE parameter:

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

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 for DFHJUP

You can use the control statements, CONTROL, OPTION, and END, to guide DFHJUP through the stages described in "Using DFHJUP to read log streams" on page 9.

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 Figure 3) 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.

```
1         10        16

CONTROL   CNTL      [{SKIP|K}={0|number}]
                    [,{STOPAFT|H}={EOF|number|(number,E)}]
                    [,{DDNAME|D}={SYSUT1|ddname}]
                    [,{DDNOUT|O}={SYSUT4|ddname}]
```

Figure 3. The DFHJUP CONTROL statement

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 embedded 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. If you do not specify this keyword, the default value of EOF is assumed.

number
must be specified in the range 0 through 9999999, with no embedded commas.

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 Figure 4) 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.

```
1         10       16

OPTION    {PRINT|  [{OFFSET|O}={1|number}]
           COPY|
           NEGOF}  [,{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]
```

Figure 4. The DFHJUP OPTION statement

Options

Each option has two distinct functions:

1. Determine the starting position for the OFFSET keyword
2. Determine the output processing to be performed.

If individual options are combined to form a multifield test, the use of OFFSET remains unchanged; however, output processing is determined by the option coded with the COND=E keyword.

PRINT
causes all selected records to be displayed on the SYSPRINT data set.

COPY
causes all selected records to be transferred to the specified output data set. You can also write these records on the SYSPRINT data set by coding the PRTSYS keyword.

NEGOF
causes the OFFSET keyword value to be used as a negative offset from the end of the journal record. All records selected using this function are displayed on the SYSPRINT data set.

All 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
To ignore the DCB information from the original data set, specify NEWDCB when using the COPY function. Supply the new DCB information on the JCL for the output data set. You can use this process to create an output data set in COMPAT41 mode from a logstream that is in the format introduced at CICS Transaction Server for OS/390, Version 1 Release 1. You can specify on the JCL that the output data set will be in variable blocked format.

Note: You may want to use NEWDCB to specify output data set attributes when your input log data has not been converted into COMPAT41 form. That is, the data is in the newer log data format introduced in CICS Transaction Server for OS/390 Version 1 Release 1. Using NEWDCB to COPY data to a BSAM data set means that the data must conform to the various access method restrictions and requirements. For example, BSAM data cannot exceed 32 KB in size. Also, data copied to a variable, or variable blocked, data set (recfm=V or recfm=VB) using NEWDCB, requires the data to be in variable format. This means that the newer log record format will cause DFHJUP to fail with an S002 if used on a copy step with NEWDCB, because the input data is not in a variable format as its records do not begin with an LLBB fullword.

If you want to use NEWDCB against input data in the format that was introduced in CICS Transaction Server for OS/390 Version 1 Release 1, you can define an exit program to DFHJUP that modifies the format of the records to put an LLBB value at their start. Doing this allows the use of NEWDCB to copy the data into a different record format.

The following example exit program can be used to achieve this result. The program modifies the first word of each record, to replace it with an LLBB field, and also avoids returning CICS block header records, as they are for internal use by CICS. Define the exit program to DFHJUP using the EXITR= or

E=OPTION statement.

```
*****************************************************************
*                                                               *
* MODULE NAME = JUPEXIT                                         *
*                                                               *
* DESCRIPTIVE NAME = Sample exit program for DFHJUP             *
*                                                               *
* @BANNER_START                          01                     *
* Licensed Materials - Property of IBM                          *
*                                                               *
*  5655-S97             JUPEXIT                                 *
*                                                               *
* (C) Copyright IBM Corp. 2009                                  *
*                                                               *
* CICS                                                          *
* (Element of CICS Transaction Server                           *
* for z/OS, Version 4 Release 2)                                *
* @BANNER_END                                                   *
*                                                               *
* STATUS = 4.2.0                                                *
*                                                               *
* Provide a sample exit routine for DFHJUP, to parse log        *
* records and reject block header records. Adjust remaining     *
* records to have an LLBB at their start. This means that       *
* the remaining CICS TS log records returned by DFHJUP are      *
* in RECFM=VB format, and so are eligible for blocking by       *
* BSAM if COPY is used with a NEWDCB for the output data set.   *
*****************************************************************
         DFHREGS                     Establish register equates
JUPEXIT  CSECT
         STM   R14,R12,12(R13)       Save the registers
         BASR  R3,0                  Establish base register
         USING *,R3                  Tell the assembler
         ICM   R4,15,0(R1)           Address record
         BZ    EOF                   If plist empty then eof
         CLC   0(0,R4),=CL4'>DFH'    Test for block hdr record
         BE    EXITREJ               If so, reject record
         L     R5,0(,R4)             Pick up record length
         SLL   R5,16                 Convert to LLBB format
         ST    R5,0(,R4)             Store back at start
*        WTO   'DFHJUP EXIT RAN'     Diagnostic message
         B     EXIT                  Leave the program
EOF      DS    0H
*        WTO   'EOF ENCOUNTERED'     Diagnostic message
EXIT     DS    0H
         LM    R14,R12,12(R13)       Restore the registers
         SR    R15,R15               Set a good return code
         BR    R14                   Return to DFHJUP
EXITREJ  DS    0H
         LM    R14,R12,12(R13)       Restore the registers
         LA    R15,1                 Set a bad rc (no base)
         BR    R14                   Return to DFHJUP
         DROP  R3                    Tell the assembler
         LTORG                       Define the literal pool
         END
```

END statement

When you have defined all tests for the current input file, use the END statement (see Figure 5 on page 18) to initiate the tests.

Positions 10 and upward can be used for comments.

```
1         10        16

END       [.....comments....]
```

Figure 5. The DFHJUP END statement

COMMENTS statement

The COMMENTS statement (see Figure 6) is optional. If used, it causes the contents to be displayed on the SYSPRINT data set.

```
1         10        16

*
```

Figure 6. The COMMENTS statement

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 Managing auxiliary storage for advice on how to define your system log.

General logs

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 23 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

- 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" on page 18, 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 Managing auxiliary storage.

Example of log data that is accessible to DFHJUP

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

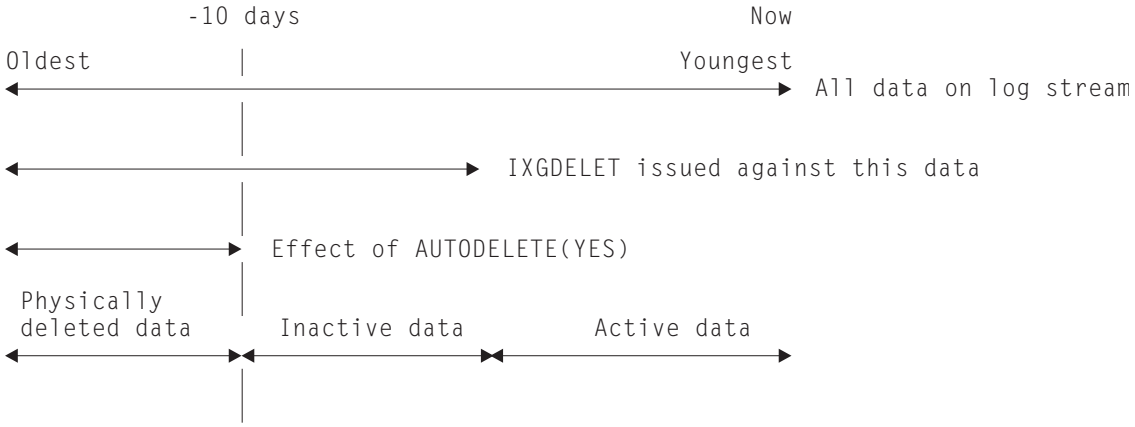


Figure 7. Active and inactive data on a log stream. The log stream has been defined with a RETPD value of 10.

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 8 on page 21.

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 8 on page 21 for an example of this diagnostic information.

```
Block identifier - 0000000000008F2A
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
                                                         *>DFH. ..IYCLZCCC..Z.......7.O...*
 000020            00000000 00000001                     *........                         *
 000000   000028   0000004C 00000038  00000014 AEEFF969   9E36E800 AEF006BE  D17EE800 C3C5C3C9
                                                         *...<.........9...Y..0..J=Y.CECI*
 000020            0000024C F8F7F3F6  0001D3C7 40404040   40404040 00000000  40F5F1F0 C9E8C3D3
                                                         *...<8736..LG        .... 510IYCL*
 000040            E9C3C3C3 E6D9C9C7  C8E3C140           *ZCCCWRIGHTA                     *
 000000   000074   00000049 00000038  00000011 AEEFF969   9ECA6000 AEF006BE  D2126000 C3C5C3C9
                                                         *.............9...-..0..K.-.CECI*
 000020            0000024C F8F7F3F6  0002E4D1 C4C6C8D1   F0F34040 00000000  0000000C C1E64040
                                                         *...<8736..UJDFHJ03  ........AW  *
 000040            00000000 E3C5E2E3  F1                 *....TEST1                       *
```

Figure 8. 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.

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

The following example shows the JCL and control statements required to print to the output data set all of 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=CICSTS54.CICS.SDFHLOAD,DISP=SHR
//SYSPRINT  DD SYSOUT=A,DCB=RECFM=FBA
//SYSUT1    DD DSNAME=CICSDA#.CICSDA1.JRNL054,
//             DCB=BLKSIZE=32760,
//             SUBSYS=(LOGR,DFHLGCNV,
//            'FROM=(1995/001,06:00),TO=(1995/007,23:59),LOCAL')
//SYSIN     DD *
*--------------------------------------------------------*
*    CONTROL STATEMENT : DEFAULTS                        *
*                INPUT = SYSUT1                          *
*               OUTPUT = SYSPRINT                        *
* SELECTION QUALIFIERS :                                 *
*           1. DEFAULT = ALL INPUT RECORDS               *
*--------------------------------------------------------*
OPTION PRINT
END
*--------------------------------------------------------*
/*
```

Figure 9. DFHJUP program, example 1

Example 2

The following example 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.

```
//JNLCOPY1 JOB (accounting information),CLASS=A
//COPYJNL   EXEC PGM=DFHJUP
//STEPLIB   DD DSNAME=CICSTS54.CICS.SDFHLOAD,DISP=SHR
//SYSPRINT  DD SYSOUT=A,DCB=RECFM=FBA
//SYSUT1    DD DSNAME=CICSAA#.CICSDC1.JRNL001,
//             DCB=BLKSIZE=32760,
//             SUBSYS=(LOGR,DFHLGCNV,'TO=(,10:00)',COMPAT41)
 //SYSUT4    DD DSNAME=EXAMPLE1.COPY1,DISP=(NEW,CATLG),
//             UNIT=SYSDA,VOL=SER=USRPAK,
//             SPACE=(TRK,(3,1))
//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,DFHLGCNV,'TO=(,10:00)',DELETE)
//CHKCOPY ENDIF
*-------------------------------------------------------*
/*
```

Figure 10. DFHJUP program, example 2

Example 3

This 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,DFHLGCNV,'TO=(1995/229,09:30)',DELETE)
```

Figure 11. IEFBR14 program, example 3. JCL and control statements to delete a log stream tail

Example 4

The following example shows the JCL and control statements required to take two copies of the same log stream partition, using the SETBRCUR and REPBRCUR options, and then to delete the log stream partition using the DELBRCUR option.

```
//ARCHJNL  JOB (accounting information),CLASS=A
//ARCHJNL1 EXEC PGM=DFHJUP
//STEPLIB   DD DSNAME=CICSTS54.CICS.SDFHLOAD,DISP=SHR
//SYSPRINT  DD SYSOUT=A,DCB=RECFM=FBA
//SYSUT1    DD DSNAME=CICSAA#.CICSDC1.JRNL001,
//             DCB=BLKSIZE=32760,
//             SUBSYS=(LOGR,DFHLGCNV,
//             'TO=(,12:00),GMT',
//             'COMPAT41,SETBRCUR')
//SYSUT4    DD DSNAME=JRNL001.ARCHIVE1,DISP=(NEW,CATLG),
//             UNIT=SYSDA,VOL=SER=USRPAK,
//             SPACE=(TRK,(3,1))
//SYSIN     DD *
*-------------------------------------------------------*
*    CONTROL STATEMENT : DEFAULTS                       *
*                INPUT = SYSUT1                         *
*               OUTPUT = SYSUT4                         *
* SELECTION QUALIFIERS :                                *
*           1. DEFAULT = ALL INPUT RECORDS              *
*-------------------------------------------------------*
OPTION COPY
END
//ARCHJNL2 IF (ARCHJNL1.RC = 0) THEN
//ARCHJNL1 EXEC PGM=DFHJUP
//STEPLIB   DD DSNAME=CICSTS54.CICS.SDFHLOAD,DISP=SHR
//SYSPRINT  DD SYSOUT=A,DCB=RECFM=FBA
//SYSUT1    DD DSNAME=CICSAA#.CICSDC1.JRNL001,
//             DCB=BLKSIZE=32760,
//             SUBSYS=(LOGR,DFHLGCNV,,REPBRCUR)
//SYSUT4    DD DSNAME=JRNL001.ARCHIVE2,DISP=(NEW,CATLG),
//             UNIT=SYSDA,VOL=SER=USRPAK,
//             SPACE=(TRK,(3,1))
//SYSIN     DD *
*-------------------------------------------------------*
*    CONTROL STATEMENT : DEFAULTS                       *
*                INPUT = SYSUT1                         *
*               OUTPUT = SYSUT4                         *
* SELECTION QUALIFIERS :                                *
*           1. DEFAULT = ALL INPUT RECORDS              *
*-------------------------------------------------------*
OPTION COPY
//ARCHJNL2 ENDIF
//JNLDEL   JOB (accounting information),CLASS=A
//IEFBR14  EXEC PGM=IEFBR14
//LOGSTRM DD DSNAME=CICSAA#.CICSDC1.JRNL001,
//           SUBSYS=(LOGR,DFHLGCNV,,DELBRCUR)
//*
```

Figure 12. DFHJUP program, example 4

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 Version 5 Release 4 format.

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.

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.

Version 5 Release 4 format

Locating records using the system-type ID field

About this task

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.

About this task

It must appear in the same form in the VALUE parameter. To do this:

Procedure

1. 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`.
2. 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.

About this task

```
//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
About this task

At Version 5 Release 4, 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.

About this task

```
//SYSIN     DD *
OPTION PRINT OFFSET=37,FLDTYP=C,VALUE=T004,FLDLEN=4,COND=E
END
/*
```

Selection using more than one search parameter

To select records using more than one search parameter, you will need more than one OPTION statement.

About this task

If you want to print all the file control records for a particular task you need 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
About this task

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 Writing a good night program.

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.

About this task

To do this:

Procedure

1. 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.
2. 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.

About this task

```
//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

About this task

The time must be entered in the form hhmmsss+ 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.

About this task

```
//SYSIN     DD *
OPTION PRINT OFFSET=27,FLDTYP=C,VALUE=T004,FLDLEN=4,COND=E
END
/*
```

Selection using more than one search parameter

To select records using more than one search parameter, you will need more than one OPTION statement.

About this task

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 2. OPTION parameter values for Version 5 Release 4 journal records. All the fields in this table are general log header fields.

Field name	OFF SET	FLD TYP	VALUE (example)	FLD LEN	Contents
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
GLRH_TRAN_ID	29	X	E3D9D5F1	4	alternative format
GLRH_TASK_ID	33	X	0000025C	4	Task Number

Table 2. OPTION parameter values for Version 5 Release 4 journal records (continued). All the fields in this table are general log header fields.

Field name	OFF SET	FLD TYP	VALUE (example)	FLD LEN	Contents
GLRH_TERM_ID	37	C	T004	4	Terminal identifier
GLRH_TERM_ID	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 3. OPTION parameter values relevant for records presented in CICS/ESA 4.1 format. In this table, the fields between offset 1 and offset 9 are system header fields. The fields between offset 11 and offset 27 are main system prefix fields.

Field name	OFFSET	FLDTYP	VALUE (example)	FLDLEN	Contents
JCRLL	1	X	0037	2	Length of record
JCRSTRID	5	X	EF59	2	System type ID
JCRSTRID	5	X	EF	1	Function identifier
JCRSTRID	6	X	59	1	Module identifier
JCRUTRID	7	X	12EF	2	User type ID
JCRLRN	9	X	002C	2	Record number within block
JCSPLL	11	X	0014	2	Length of system prefix
JCSPTASK	16	X	00025C	3	Task number
JCSPTIME	19	X	1445123F	4	Time of request - hhmmsss+
JCSPTRAN	23	C	TRN1	4	Trans-action identifier
JCSPTRAN	23	X	E3D9D5F1	4	alter-native format
JCSPTERM	27	C	T004	4	Terminal identifier
JCSPTERM	27	X	E3F0F0F4	4	alternative format

Chapter 2. 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 `ON` for the STATRCD system initialization parameter. The other statistics record types (unsolicited, requested and end-of-day) are written regardless of the setting of the **STATRCD** parameter. For information about the SMF data sets, see z/OS MVS System Management Facilities (SMF). For information about what CICS data is recorded on the SMF data sets, see Introduction to CICS statistics in Monitoring. For information about interpreting CICS statistics output in the DFHSTUP report, see CICS statistics in DSECTS and DFHSTUP report in Monitoring.

Always use the version of the DFHSTUP program from the same release of CICS as the data that it is to process. This section describes the CICS TS for z/OS®, Version 5.4 version of the DFHSTUP program, which you can use only for CICS TS for z/OS, Version 5.4 data.

Setting the statistics recording status, end-of-day time and recording interval

The statistics recording status is set at CICS startup by the **STATRCD** system initialization parameter. The end-of-day time is set by the **STATEOD** system initialization parameter, and the recording interval is set by the **STATINT** system initialization parameter.

About this task

You can modify all these values by using CEMT SET STATISTICS or the **SET STATISTICS** command. See and SET STATISTICS.

In the CICS Explorer, the Regions view provides a functional equivalent to this command.

The system initialization parameters can be used as follows:

STATEOD

Specifies the end-of-day time in the format `hhmmss`. The default is 0, which is midnight.

End-of-day time is expressed in local time and must be in the range 00:00:00-23:59:59. That is, the `hh` value cannot exceed 23, and the `mm` and `ss` values can be specified in the range 00 to 59. If you leave out leading zeros, the DFHSIT macro inserts them. For example, 100 becomes 000100; that is, 1 minute 00 seconds past midnight.

This system initialization parameter is equivalent to the ENDOFDAY option on the CEMT SET STATISTICS and the **SET STATISTICS** command. You can use this option to modify the value set by **STATEOD**.

© Copyright IBM Corp. 1989, 2017

STATINT

Specifies the recording interval for system statistics in the format `hhmmss`. The default is 1 hour.

The interval must be at least one minute and cannot be more than 24 hours. The minutes and seconds part of the value can be specified in the range 00 to 59. If you leave out leading zeros, the DFHSIT macro inserts them. For example, 3000 becomes 003000; that is, an interval of 30 minutes.

This system initialization parameter is equivalent to the INTERVAL option on the CEMT SET STATISTICS and the **SET STATISTICS** command. You can use this option to modify the value set by **STATINT**.

STATRCD

Specifies whether interval statistics are collected. Values can be as follows:

OFF (default)
Interval statistics are not collected.

End-of-day, unsolicited, and requested statistics are written to SMF regardless of the STATRCD setting. End-of-day statistics are collected at the logical end of day and on shutdown. Unsolicited statistics are written to SMF when resources are discarded or closed.

ON Interval statistics are collected.

On a cold start of a CICS region, interval statistics are recorded by default at hourly intervals. All intervals are timed using the end-of-day time as a base starting time (not CICS startup time). The default end-of-day time is midnight, so the default settings result in collections at 00.00, 01.00, 02.00, 03.00, and so on, regardless of the time that you start CICS.

You can change the statistics recording status at any time in the following ways:

- During a warm or emergency restart, by coding the **STATRCD** system initialization parameter.
- While CICS is running, by using CEMT SET STATISTICS or the **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 CEMT PERFORM STATISTICS or the **PERFORM STATISTICS** command. See and PERFORM STATISTICS.

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 one of the following:

- The beginning of the current interval.
- The logical end-of-day collection time.
- The time that you last issued a SET or PERFORM STATISTICS command specifying the RESETNOW option.

"Job to run the DFHSTUP program" on page 33 gives information about how to use the DFHSTUP program to select and format CICS statistics.

Support for Katakana 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

Sample JCL to unload the SMF data sets and run DFHSTUP against the data.

The job, which is shown in Figure 13 on page 34 comprises two job steps. The job steps are as follows:

1. Unload the SMF data set or data sets containing the CICS statistics that you want to process.
2. Run the statistics utility program to sort, format, and print the statistics data. You run the DFHSTUP program in a batch region to process any CICS SMF type 110 statistics records that are present in an unloaded SMF data set, which you can write to either a temporary or a cataloged data set.

```
//STUP     JOB 'accounting info',CLASS=A,
//             USER=userid,MSGCLASS=A,NOTIFY=userid
//*********************************************************************
//*  Step 1: Unload data from the SMF data sets
//*********************************************************************
//SMFDUMP  EXEC PGM=IFASMFDP
//INDD1     DD DSN=SYS1.MANx,DISP=SHR,AMP=('BUFSP=65536')            1
//INDD2     DD DSN=SYS1.MANy,DISP=SHR
//OUTDD1    DD DSN=user.SMF.DATA,DISP=(NEW,CATLG),                   2
//             SPACE=(CYL,(50,10)),UNIT=SYSDA
//SYSPRINT  DD SYSOUT=A
//SYSIN     DD *
       INDD(INDD1,OPTIONS(DUMP))                                    1
       INDD(INDD2,OPTIONS(DUMP))
       OUTDD(OUTDD1,TYPE(0:255))                                    2

//*********************************************************************
//*  Step 2: Sort, format and print the statistics records           3
//*********************************************************************
//STUP1    EXEC PGM=DFHSTUP,REGION=0M
//*****************************************
//STEPLIB  DD DSN=CICSTS54.CICS.SDFHLOAD,DISP=SHR
//         DD DSN=CICSTS54.CICS.SDFHAUTH,DISP=SHR
//DFHSTATS  DD DSN=user.SMF.DATA,DISP=SHR                           4
//DFHSTWRK  DD UNIT=SYSDA,SPACE=(CYL,(8,4))                         5
//SORTWK01  DD UNIT=SYSDA,SPACE=(CYL,(4))                           6
//SORTWK02  DD UNIT=SYSDA,SPACE=(CYL,(4))
//SORTWK03  DD UNIT=SYSDA,SPACE=(CYL,(4))
//SORTWK04  DD UNIT=SYSDA,SPACE=(CYL,(4))
//DFHPRINT  DD SYSOUT=*                                             7
//SYSPRINT  DD SYSOUT=*
//SYSUDUMP  DD SYSOUT=*
//SYSABEND  DD SYSOUT=*
//SYSOUT    DD SYSOUT=*
//SYSIN     DD *                                                    8
SELECT APPLID=(applid1,applid2)
COLLECTION TYPE=ALL
/*
//
```

Figure 13. Example job to extract and print statistics data

Note:

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 13, 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. The DFHSTUP program always produces a report in chronological order.

   For more information about unloading SMF data sets, see the*z/OS MVS System Management Facilities (SMF)* .

   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 Figure 27 on page 87.)
2. The OUTDD parameter refers to an OUTDD1 DD statement that defines 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 processes only 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 DFHSTUP program sorts statistics records in the sequence: specific applid, date (in YYYYMMDD form), and time.
4. The ddname for the input to the DFHSTUP program must be DFHSTATS. This input is the unloaded SMF data set.
5. The ddname for the DFHSTUP work file must be DFHSTWRK. Fve types of statistics records can be written to the DFHSTWRK data set:
   - Files
   - Log streams
   - Transactions
   - Transient data queues
   - DB2® Entries

   The size of the DFHSTWRK data set required depends on the largest set of resources from this list being written to the data set.

   The following calculation can be used to estimate the size of the DFHSTWRK data set required:
   - Files
     a. The length of the file statistics dsect, DFHA17DS, is 376 bytes.
     b. The amount of file data that is written in one interval/requested reset/End-of-day report is 376 * (number of files in region) bytes.
     c. Because file statistics are written to SMF whenever a file is closed online, add an estimate of the number of file closes, multiplied by the length of the DFHA17DS dsect (376 bytes).
   - Log streams
     a. The length of the log streams statistics dsect, DFHLGSDS, is 124 bytes.
     b. The amount of log stream data that is written in one interval/requested reset/End-of-day report is 124 * (number of log streams in region) bytes.
   - Transactions
     a. The length of the transaction statistics dsect, DFHXMRDS, is 164 bytes.
     b. The amount of transaction data that is written in one interval/requested reset/End-of-day report is 164 * (number of transactions in region) bytes.
   - Transient data queues
     a. The length of the transient data queue statistics dsect, DFHTQRDS, is 188 bytes.
     b. The amount of transient data queue data that is written in one interval/requested reset/End-of-day report is 188 * (number of transient data queues in region) bytes.
     c. Additionally, for extrapartition transient data queues, a record is written when the transient data queue is closed.
   - DB2 Entries
     a. The length of the DB2 entries dsect, DFHD2RDS, is 204 bytes.
     b. The amount of DB2 entry data that is written in one interval/requested reset/End-of-day report is 204 * (number of DB2 entries in region) bytes.
     c. Additionally, a record for each DB2 entry is written when the DB2 connection is closed.

6. The DFHSTUP program sorts the data by using 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.
7. The ddname for the output from the DFHSTUP program must be DFHPRINT, which you can direct either to a data set or printer.
8. The control parameters for the DFHSTUP program can be supplied only in the SYSIN data set.

   Each control parameter in the SYSIN data set starts 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 - 71 and that 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" on page 42.

The DFHSTUP extract statistics reporting function

The extract reporting facility provides a method of sending CICS statistics data to a user program that can process statistics records to create tailored reports.

Make these reports easy to review to avoid working through large amounts of data to determine if some corrective or preventive tuning action is required. You can identify the specific CICS regions, the time of day, and the type of CICS resources that might require further specific in-depth performance analysis. DFH0STXR and DFH0STXD are sample programs that are designed to use the extract reporting function. You can use these sample programs as supplied, or as a model on which to base your own programs.

The DFH0STXR sample program

The DFH0STXR sample program produces an event type of statistics report that highlights each event in a single print line and provides a convenient way to analyze CICS statistics records.

The types of event are as follows:

- Exception. A CICS limit condition has been exceeded (for example MXT, or file strings).
- Warning. A peak condition has met or exceeded a user-defined threshold percentage.
- Information. A peak value has been reached for a key CICS system parameter.

These events are predefined in the DFH0STXR sample program but can be customized (see "Using the extract statistics reporting function" on page 39 for guidance). To help with customization, events and the statistics fields queried for each event are documented in the prolog.

The existing statistics utility program record selection capabilities, such as *applid*, *date* and *time*, apply automatically to the DFH0STXR reports, in addition to any other statistics reports that might be created at the same time. However, the COLLECTION TYPE and UPPERCASE control parameters are ignored.

You have the option in the design of the user program to write the statistics data to a data set to be used as input to a database, such as DB2, or to a reporting tool such as CICS Performance Analyzer (CICS PA) for further analysis.

The source code for DFH0STXR is supplied in theCICSTS54.CICS.SDFHSAMP library. It is also supplied in compiled form in CICSTS54.CICS.SDFHLOAD. See Installing application programs for guidance on installing application programs.

As well as the DFH0STXR sample, two skeleton exits are provided; an assembler extract exit called DFH£STXA, and a COBOL extract exit called DFH0STXC. These show the format and structure of the interface between DFHSTUP and the extract exit. The source for these skeletons is in SDFHSAMP, and the compiled programs in SDFHLOAD. Extract exits must be written using a Language Environment® enabled assembler or a CICS Language Environment supported high level language.

Figure 14 on page 38 contains sample output from the DFHSTUP statistics utility program including output from the extract exit. Other statistics reports have been suppressed, as reported in message DFHST0233I.

```
CICS 6.3.0 Statistics Utility Program                              Report Date 08/30/2002   Report Time 15:51:18   Page     4
Report Abstract
_____________________________________________________________________________________________________________________________
REPORT ABSTRACT
_______________
  Total SMF records read . . . . . . . . . . . :                  1315
  Total SMF/CICS records read. . . . . . . . . :                  1313
  Total SMF/CICS/Statistics records read . . . :                  1313
  Total SMF/CICS/Statistics records processed. :                    35
  Applid: IYK2Z2G1  Jobname: CI13JTD5

  ___________________________________
  Number of Statistics Intervals processed . . :                     0
  Number of EOD records processed. . . . . . . :                     0
  Number of INT records processed. . . . . . . :                     0
  Number of REQ records processed. . . . . . . :                  2456
  Number of RRT records processed. . . . . . . :                     0
  Number of USS records processed. . . . . . . :                    15
  Date/Time of first SMF record processed. . . : 08/30/2002 at 15:40:25     CICS Start Date/Time. . : 08/30/2002 at 15:40:23
  Date/Time of last SMF record processed . . . : 08/30/2002 at 15:42:43
DFHST0233 I DFHSTUP report suppressed. Report options requested only an EXTRACT userprogram to be invoked.
CICS 6.3.0 Statistics Utility Program                              Report Date 08/30/2002   Report Time 15:51:18   Page     5
Report Abstract
_____________________________________________________________________________________________________________________________
CICS 6.3.0 Statistics Exception Event Reporter                     Report Date: 08/30/2002   Report Time: 15:51:18      Page      1
_____________________________________________________________________________________________________________________________
EXCEPTION
WARN
INFO
CICS 6.3.0 Statistics Exception Event Reporter                     Report Date: 08/30/2002   Report Time: 15:51:18      Page      2
_____________________________________________________________________________________________________________________________
        Record     Record  Stats           Event
Type     Date       Time    Type  Applid    id    Event Description
_____________________________________________________________________________________________________________________________
  I   08/30/2002  15:42:43  REQ  IYK2Z2G1  XM001  MXT Limit =  30. Current Tasks =   1. Peak Tasks =   6.
  I   08/30/2002  15:42:43  REQ  IYK2Z2G1  SM001  DSA Limit =      4,096K. Allocated =      2,048K. Peak =      2,048K.
  I   08/30/2002  15:42:43  REQ  IYK2Z2G1  SM002  EDSA Limit =    204,800K. Allocated =     22,528K. Peak =     22,528K.
  I   08/30/2002  15:42:43  REQ  IYK2Z2G1  TS001  DFHTEMP CIs =        507. Peak Used =          2. Peak TSMain =          0K.
  W   08/30/2002  15:42:43  REQ  IYK2Z2G1  LD001  Program Load to Use Ratio for Program 'IBMRLIB1' = 100%. Location = 'RDSA    '.
  W   08/30/2002  15:42:43  REQ  IYK2Z2G1  LD001  Program Load to Use Ratio for Program 'IBMRSAP ' = 100%. Location = 'RDSA    '.
  W   08/30/2002  15:42:43  REQ  IYK2Z2G1  LD001  Program Load to Use Ratio for Program 'IGZCPAC ' = 100%. Location = 'ERDSA   '.
  W   08/30/2002  15:42:43  REQ  IYK2Z2G1  LD001  Program Load to Use Ratio for Program 'IGZCPCC ' = 100%. Location = 'ERDSA   '.
  I   08/30/2002  15:42:43  REQ  IYK2Z2G1  D2001  DB2 Conn 'RCTJT   '. TCB Limit =    12. Peak TCBs =     1. TCB Readyq HWM =   0.
CICS 6.3.0 Statistics Exception Event Reporter                     Report Date: 08/30/2002   Report Time: 15:51:18      Page      3
_____________________________________________________________________________________________________________________________
                                  Statistics Exception Event Report Totals for Applid IYK2Z2G1
                                  ___________________________________________________________
Statistics Records Processed . . . . . . . . . . :       2,471
Statistics Exception Events Reported . . . . . . :           0
Statistics Warning Events Reported . . . . . . . :           4
Statistics Information Events Reported . . . . . :           5
                                                      _________
Total Events Reported  . . . . . . . . . . . . . :           9
_____________________________________________________________________________________________________________________________
Total Statistics Records Processed .................:      2,471
Statistics Event Report Processing Complete.
```

Note: The date and time do not indicate when an event occurs. The statistics record might be written out some time after the actual event has occurred.

Figure 14. Sample output from DFHSTUP using the DFH0STXR sample extract program

The DFH0STXD sample program

The DFH0STXD sample extract program produces a basic report from the CICS statistics records for installed CICS resources. Each print line displays details for the resource listed including the resource type, the define source and the installation signature.

The DFH0STXD sample program can be customized (see “Using the extract statistics reporting function” on page 39 for guidance).

The existing record selection capabilities of the statistics utility program, such as, *applid*, *date* and *time*, apply automatically to the DFH0STXD reports, in addition to the installed resource statistics reports.

You have the option in the design of the user program to write the statistics data to a data set to be used as input to a database, such as DB2, or to a reporting tool such as z/OS DFSORT's ICETOOL utility.

The source code for DFH0STXD is supplied in the CICSTS54.CICS.SDFHSAMP library. It is also supplied in compiled form in CICSTS54.CICS.SDFHLOAD.

The JCL and DFHSTUP control statements required to invoke DFH0STXD are:

```
//STXDRPT  DD SYSOUT=x
EXTRACT USERPROGRAM=DFH0STXD
```

The following example shows these control statements:

```
//xxxxxxxx JOB (job accounting)
//stepname EXEC PGM=DFHSTUP,REGION=0M
//STEPLIB  DD DSN=....
//         DD DSN=....
//DFHSTATS DD DSN=....
//SYSPRINT DD SYSOUT=x
//SYSOUT   DD SYSOUT=x
//STXDRPT  DD SYSOUT=x
//DFHPRINT DD SYSOUT=x
//SYSIN DD *
SELECT APPLID=(xxxxxxxx)
SELECT TYPE=(yyyyyyy)
EXTRACT USERPROGRAM=DFH0STXD
COLLECTION TYPE=ALL
SUMMARY
/*
//
```

A sample extract from the DFH0STXD report using the following control statements is shown here:

```
SELECT APPLID=(IYK0AS)
SELECT TYPE=(LIBRARY)
EXTRACT USERPROGRAM=DFH0STXD
COLLECTION TYPE=ALL
SUMMARY
```

```
CICS 6.6.0 Statistics Resource Definition Reporter                    Report Date 03/31/2008   Report Time 14:56:52   Page     1

 <- Statistics Record  ->          Resource                                  Define    <---------- Install Signature ---------->
    Date      Time   Type Applid     Type       Resource ID                  Source    Agent           Date      Time     Userid

 03/31/2008 14:21:44  REQ IYK0AS   LIBRARY      DFHRPL                       SYSTEM    SYSTEM       03/31/2008 12:03:48  ASTEWAR
 03/31/2008 14:21:44  REQ IYK0AS   LIBRARY      BOB                          NEWTEST   GRPLIST      03/31/2008 12:03:52  CICSUSER
 03/31/2008 14:21:44  REQ IYK0AS   LIBRARY      ALLIB                        LIBGRP    GRPLIST      03/31/2008 12:03:52  CICSUSER
 03/31/2008 14:21:44  REQ IYK0AS   LIBRARY      BIGBANG                      LIBGRP    GRPLIST      03/31/2008 12:03:52  CICSUSER
 03/31/2008 14:21:44  REQ IYK0AS   LIBRARY      ALIB1                        TESTLIB   GRPLIST      03/31/2008 12:03:52  CICSUSER
 03/31/2008 14:21:44  REQ IYK0AS   LIBRARY      ALIB11                       TESTLIB   GRPLIST      03/31/2008 12:03:52  CICSUSER
 03/31/2008 14:21:44  REQ IYK0AS   LIBRARY      ALIB12                       TESTLIB   GRPLIST      03/31/2008 12:03:52  CICSUSER
 03/31/2008 14:21:44  REQ IYK0AS   LIBRARY      ALIB2                        TESTLIB   GRPLIST      03/31/2008 12:03:52  CICSUSER
 03/31/2008 14:21:44  REQ IYK0AS   LIBRARY      BLIB1                        TESTLIB   GRPLIST      03/31/2008 12:03:52  CICSUSER
 03/31/2008 14:21:44  REQ IYK0AS   LIBRARY      BLIB11                       TESTLIB   GRPLIST      03/31/2008 12:03:52  CICSUSER

CICS 6.6.0 Statistics Resource Definition Reporter                    Report Date 03/31/2008   Report Time 14:56:52   Page     2
```

Figure 15. Sample output from DFHSTUP using the DFH0STXD sample extract program

Using the extract statistics reporting function

About this task

The existing statistics utility program record selection capabilities, such as, *applid*, *date* and *time* apply automatically to extract reports as well as to any other statistics

reports that may be created at the same time. However, the COLLECTION TYPE control parameter of DFHSTUP is ignored in an EXTRACT report.

Important: You must use the JCL statements in this topic with the full DFHSTUP program job in the topic "Job to run the DFHSTUP program" on page 33.

To use the extract reporting function:

Procedure

1. Specify the EXTRACT control statement to control the output from DFHSTUP. The EXTRACT control statement takes all the statistics data from the input SMF data set and passes it to a user program for processing. The EXTRACT control statement has the following syntax:

   ```
   EXTRACT USERPROGRAM=progname
   ```

   where USERPROGRAM is the name of the user-written program that is to process the data. You must supply a USERPROGRAM value.

   Specifying the EXTRACT control statement changes the default DFHSTUP report settings. If you specify only the EXTRACT control statement, only the extract exit is driven; other DFHSTUP reports are suppressed. If EXTRACT is specified, other statistics report control statements, such as SUMMARY, must also be specified to ensure that the appropriate reports are produced.
2. Use the STXRINPT DD control statement to specify the event types to be reported. The possible event types are:

   ```
   [EXCEPTION] [WARNING] [INFORMATION]
   ```

   Exception is the default. You can abbreviate these parameters to EXC, WARN and INFO.
3. If you want to print all types of statistics reports, including a summary and an event report, you could specify the following:

   ```
   //xxxxxxxx JOB (job accounting)
   //stepname EXEC PGM=DFHSTUP,REGION=0M
   //STEPLIB  DD DSN=....
   //         DD DSN=....
   //         DD DSN=PP.CEE.SCEERUN,DISP=SHR
   //DFHSTATS DD DSN=...
   //DFHSTWRK DD ....
   //SYSPRINT DD SYSOUT=x
   //STXRRPT  DD SYSOUT=x
   //STXRINPT DD *
   INFO
   /*
   //SYSIN DD *
   COLLECTION TYPE=ALL
   SUMMARY
   EXTRACT USERPROGRAM=DFH0STXR
   /*
   //
   ```
4. To produce an event-type report using the CICS supplied sample exit program, you could specify the following:

   ```
   //xxxxxxxx JOB (job accounting)
   //stepname EXEC PGM=DFHSTUP,REGION=0M
   //STEPLIB  DD DSN=....
   //         DD DSN=....
   //         DD DSN=PP.CEE.SCEERUN,DISP=SHR
   //DFHSTATS DD DSN=...
   //DFHSTWRK DD ....
   //SYSPRINT DD SYSOUT=x
   ```

```
//STXRRPT  DD SYSOUT=x
//STXRINPT DD *
INFO                          <- Optional input
/*
//SYSIN DD *
EXTRACT USERPROGRAM=DFH0STXR
/*
//
```

5. Assemble (or compile) and link edit DFHSTUP user programs as batch programs, not as CICS applications.
6. Link-edit control statements appropriate to the language in which they are written. When you link-edit the programs, you need to specify the following link-edit control statements:
   - An ENTRY statement that defines the entry name as DFHEXTRA
   - An INCLUDE statement for a CICS-supplied stub that must be included in your user program
   - A CHANGE statement to change the dummy CSECT name in the CICS-supplied stub from EXITEP to the name of you user program.

   These requirements are illustrated in the following sample job stream.

```
//DFHxxxxx JOB (),CLASS=A,MSGCLASS=A,NOTIFY=userid
//*
//* Compile job step here
//*
//LINK     EXEC PGM=IEWL,PARM='XREF,LIST,LET'
//SYSLIB   DD DSN=PP.CEE.SCEELKED,DISP=SHR
//CICSLIB  DD DSN=CICSTS54.CICS.SDFHLOAD,DISP=SHR
//OBJLIB   DD DSN=object.module.library,DISP=SHR
//SYSLMOD  DD DSN=user.library,DISP=SHR
//SYSUT1   DD UNIT=SYSDA,SPACE=(1024,(100,10))
//SYSPRINT DD SYSOUT=x
//SYSLIN   DD *
 ENTRY   DFHEXTRA
 CHANGE  EXITEP(program name)
 INCLUDE CICSLIB(DFHSTXLE)
 INCLUDE OBJLIB(obj-prog)
 MODE AMODE(31),RMODE(24)
 NAME    progname(R)
/*
//
```

   Note:

   a. The entry name, DFHEXTRA, is the entry name in the CICS-supplied stub, DFHSTXLE.
   b. The CICS-supplied stub, DFHSTXLE, is generated with a link to the user program using a dummy CSECT name (EXITEP).
   c. Use the link edit CHANGE statement to change the CSECT name from EXITEP to the name specified as the program ID or CSECT name in the user program.
   d. Include DFHSTXLE in any Language Environment-conforming user program that you write for use with the **DFHSTUP EXTRACT** command. DFHSTXLE is the interface stub between DFHSTOT, a module in DFHSTUP, and the Language Environment user program.
   e. `obj-prog` is the name of the object program.
   f. `progname` is the name for the load module; this is the name that you specify on the **USERPROGRAM** parameter of the EXTRACT command.
   g. You must link edit user programs AMODE(31) and RMODE(24).

Control parameters of the DFHSTUP program

You can use the parameters described here to control the output of statistics by the DFHSTUP program.

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.

For information about the job that you use to run the DFHSTUP program, see "Job to run the DFHSTUP program" on page 33.

The DFHSTUP reports list statistics records in the following type order:

- Statistics domain
- Transaction manager
- Transaction class
- Dispatcher
- MVS TCBs (not in summary report)
- Recovery Manager
- Enqueue Manager
- Monitoring
- Storage Manager DSA
- Storage Manager task subpool
- Storage Manager domain subpool
- Loader
- LIBRARY resources (not in summary report)
- Temporary storage
- Transient data
- VTAM® (now z/OS Communications Server)
- Terminal Autoinstall
- Program Autoinstall
- System dump
- Transaction dump
- Table manager
- Transaction
- Program definitions
- Program
- File
- LSRpool
- LSRpool file
- Transient data queue
- Journalname
- Logger
- Logstream
- WebSphere® MQ connection
- IBM MQ monitor

- DB2 connection
- DB2 entries
- TCP/IP
- TCP/IP services
- ASYNCSERVICE resources
- IPCONN resources
- BUNDLE resources
- URIMAP resources
- PIPELINE resources
- WEBSERVICE resources
- ATOMSERVICE resources
- XMTRANSFORM resources
- DOCTEMPLATE resources
- Event processing
- EVENTBINDING resources
- CAPTURESPEC resources
- JVMSERVER resources
- JVMPROGRAM resources
- Terminal
- ISC/IRC system and mode entry
- ISC/IRC attach time security
- DBCTL
- FEPI pool
- FEPI connection
- FEPI target
- User domain

For resource types that are supported as private resources for applications deployed on platforms, statistics for both public and private resources are reported. The statistics for public resources of that type appear first, followed by the statistics for private resources of that type.

Parameters

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. This parameter is ignored in an EXTRACT report.

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 2000 APPLIDs. If you specify two or more APPLIDs, you must enclose them in parentheses, and separate them by commas. If you specify more than 2000 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 2000 APPLIDs. If you specify two or more APPLIDs, you must enclose them in parentheses, and separate them by commas. If you specify more than 2000 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.

The resource types that you can code on the SELECT TYPE parameter are shown in the following list:

- ASYNCSERVICE
- ATOMSERVICE
- AUTOINSTALL
- BUNDLE
- CAPTURESPEC
- CONNECTION
- DBCTL
- DB2
- DB2CONN (a subset of DB2, including DB2 Connection statistics but omitting DB2 Entry statistics)
- DISPATCHER
- DOCTEMPLATE
- ENQUEUE
- EPADAPTER
- EVENTBINDING
- EVENTPROCESS
- FEPI
- FILE

- JOURNAL
- JVMPROGRAM
- JVMSERVER
- LIBRARY
- LOADER (a subset of PROGRAM, including Loader Global statistics but omitting Program statistics)
- LOGSTREAM
- LSRPOOL
- MONITOR
- MQCONN
- MQMONITOR
- MVSTCB
- OVERVIEW (a summary comprising Transaction Manager Global statistics, Dispatcher Global statistics, Loader Global statistics, and Storage Manager DSA statistics)
- PIPELINE
- PROGAUTO
- PROGRAM
- PROGRAMDEF
- STATS
- STORAGE
- SYSDUMP
- TABLEMGR
- TCPIP
- TCPIPSERV
- TDQUEUE
- TERMINAL
- TRANCLASS or TCLASS
- TRANDATA (a subset of TDQUEUE, including Transient Data Global statistics but omitting Transient Data Resource (Queue) statistics)
- TRANDUMP
- TRANMGR (a subset of TRANSACTION, including Transaction Manager Global statistics but omitting Transaction Manager Resource statistics)
- TRANSACTION
- TSQUEUE
- URIMAP
- USER
- VTAM (now z/OS Communications Server)
- WEBSERVICE
- XMLTRANSFORM

For resource types that are supported as private resources for applications deployed on platforms, statistics for both public and private resources are reported when you select a resource type. The statistics for public resources of that type appear first in the report, followed by the statistics for private resources of that type.

When you use SELECT TYPE, if you want to reduce the size of your statistics reports you can code the summary resource types DB2CONN, LOADER, TRANMGR, and TRANDATA instead of the complete resource types DB2, PROGRAM, TDQUEUE, and TRANSACTION. To get an overview statistics report, comprising Transaction Manager Global statistics, Dispatcher Global statistics, Loader Global statistics, and Storage Manager statistics, code the OVERVIEW resource type.

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. When you use IGNORE TYPE, you can code most of the resource types listed above for the SELECT TYPE parameter, *except* for the OVERVIEW resource type and the summary resource types (DB2CONN, LOADER, TRANMGR, and TRANDATA). The IGNORE TYPE parameter can only be used to ignore complete resource types; it does not ignore resource types that are subsets or combinations of other resource types.

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 collections, for all the specified APPLIDs. This is normally the default. However, if you specify the EXTRACT control parameter but not COLLECTION TYPE, only the extract exit is invoked and no other statistics output is produced.

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 another interval collection, being the final collection of statistics for the last interval at the time of shutdown or logical end-of-day. Because the specified interval over which interval statistics are being collected might 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. Unsolicited statistics are collected by CICS for dynamically allocated and deallocated resources. For a complete description of the unsolicited statistics record types, see CICS statistics in Monitoring.

EXTRACT USERPROGRAM=*progname*

Specifies a user program that can process statistics records to create reports of statistics data allowing you to produce tailored statistics reports.

PROGNAME

The name of the user-written program that is to process the data retrieved by the EXTRACT command. You must supply a *PROGNAME* value.

Specifying the EXTRACT exit command changes the default DFHSTUP report settings. If you specify only the EXTRACT exit, only the extract exit is driven; other DFHSTUP reports are suppressed. If the EXTRACT control statement is specified, other statistics report control statements, such as SUMMARY, must also be specified to ensure that the appropriate reports are produced. See "The DFHSTUP extract statistics reporting function" on page 36 for guidance.

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, the DFHSTUP program processes every statistics record between the DATE START and TIME START until the DATE STOP and TIME STOP.

DAILY

If DAILY is coded, the DFHSTUP program processes 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:

1. If the specified period (START time to STOP time) spans across midnight, you must also code the DATE parameter.
2. If interval statistics are not collected and written to the SMF data set during the specified period (START time to STOP time), the DFHSTUP reports might not show the information you are looking for. To see a specific item of information, ensure that your specified period covers a statistics collection that includes the item you want.

   For example, you might want to see statistics relating to a transaction that was run at 10:15. If your recording interval is set at the default of 1 hour, interval statistics are written to the SMF data set at 10:00, and again at 11:00. If you run the DFHSTUP program and specify a START time of 09:30 and a STOP time of 10:30, the DFHSTUP reports will not show the statistics relating to the transaction that was run at 10:15, because the SMF data set

during that period contains only the interval statistics that were collected at 10:00. The statistics relating to the transaction that was run at 10:15 are written to the SMF data set at the next statistics collection, at 11:00.

The times of statistics collections are controlled by the **STATINT** and **STATEOD** system initialization parameters. You can also perform a statistics collection at any time by using the CEMT or EXEC CICS command **PERFORM STATISTICS RECORD**. For example, to avoid waiting until 11.00 to see statistics relating to a transaction that was run at 10:15, you can use the **PERFORM STATISTICS RECORD** command to record statistics immediately, and then run the DFHSTUP program, specifying a period that includes the time when you entered the command.

Examples

1. To process every statistics record written between 10th September 2011 at 10:00 hours and 16th September 2011 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=09/10/2011,STOP=09/16/2011
2. To process every statistics record written between 10:00 hours and 20:00 hours each day starting on 10th September 2011 and stopping on 16th September 2011, you can code the following TIME and DATE control statements:

   TIME START=10.00.00,STOP=20.00.00,DAILY

   DATE START=09/10/2011,STOP=09/16/2011

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 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 twenty-first century
- *yyyy* = year

If yy is coded the twenty-first century is assumed. For example, a date of 12/20/16 is for the 20th December 2016 and a date of 12/20/1996 is for the 20th December 1996.

PAGE SIZE=*number*
Specifies the number of lines to be formatted per page, in the range 20 - 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 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.

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) contain 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 reduce the output significantly if you code this parameter and omit the COLLECTION TYPE parameter.

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

Chapter 3. 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, IBM 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 data sets, 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
```

© Copyright IBM Corp. 1989, 2017

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 (rollbacked) 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 16, 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 UOWs 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 16. Statistics sample program, DFH$STER

For further information about the shunting of UOWs, see Units of work .

Installing DFH$STER

About this task

Define this program in the second phase PLTPI for your CICS region. For more information, see Writing initialization and shutdown programs.

Chapter 4. Trace utility print program (DFHTU710)

There are a number of 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.
4. The JVM server trace file in z/OS Unix System Services

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 section describes how you can print the CICS region trace data from:

- The CICS auxiliary trace data sets, using the CICS trace utility program, DFHTU710.
- The GTF data sets, using a CICS-supplied routine with the MVS interactive problem control system (IPCS).

The CICS trace utility program, DFHTU710

The CICS utility program, DFHTU710, 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=DFHTU710 statement
- The DFHAXPRM data set.

You can specify that all entries are to be processed, or select specific entries for processing. For example, you might specify entries with the following characteristics:

- Entries written to the auxiliary trace data set within a specified period of time.
- Entries written for a specified terminal.
- Entries with a specified trace identifier.
- Entries with specified trace entry sequence numbers. The sequence number is given in each trace entry, and can be determined from a summary trace point.
- Entries associated with a specified transaction identifier.
- Entries associated with a specific instance of a transaction identifier (task)
- Entries associated with a selected kernel task.
- Entries 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.

© Copyright IBM Corp. 1989, 2017

For abbreviated and full trace formatting, the trace summary table at the end of the formatted output provides summary information about the trace entries that relate to each task. The trace summary table is not produced for short-format trace.

You can use the job control statements shown in Figure 17 to invoke the utility program for each auxiliary trace data set. Only use the trace utility program to print auxiliary trace data sets that you have opened in the most recent run of CICS. If you did not open an auxiliary trace data set during the most recent run of CICS, the trace utility program either prints records from a previous run or cannot recognize the records. If you opened the auxiliary trace data set A in the most recent run of CICS, but did not open auxiliary data set B, you can print data set A, but not print B.

```
//PRTRACE  JOB  accounting info,name,MSGLEVEL=1,CLASS=A,MSGCLASS=A,
//              REGION=2M                                          1
//PRINT    EXEC PGM=DFHTU710
//STEPLIB  DD   DSN=CICSTS54.CICS.SDFHLOAD,DISP=SHR
//         DD   DSN=CICSTS54.CICS.SDFHLINK,DISP=SHR
//DFHAUXT  DD   DSN=CICSTS54.CICS.DFHxxxx,DISP=SHR                 2
  3
//DFHAXPRT DD   SYSOUT=A
//DFHAXPRM DD   *
 •
 •
 •
  [trace selection parameters]                                     4
/*
```

Figure 17. Sample JCL to print CICS trace data from an auxiliary trace data set

1 The sample JCL gives a region size of 2 MB that you might typically need to run the DFHTU710 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 DFHTU710 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=CICSTS54.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. For more information, see "The trace selection parameters for DFHTU710."

The trace selection parameters for DFHTU710

Use 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. You code the trace selection parameters in the DFHAXPRM DD statement or in the PARM parameter.

For example:

```
//PRINT  EXEC PGM=DFHTU710,PARM='selection_parameter,selection_parameter,...'
```

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*).

Note: Use alternative trace selection parameters if there are more than *999999* trace entries, and, you require any entry beyond the first *999999* to be printed. Only the first set of entries with numbers from *000001* to *999999* will be printed.

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={0.0128|number of seconds}
Specifies the interval between auxiliary trace entries after which entries are 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 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 0.0128 seconds applies.

You can specify interval values in the range zero seconds (all trace entries are highlighted) through 99.9999999999 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.

LAST_BLOCKS=*n*
For internal trace, specifies that only the last *n* blocks of the trace table are printed, counting from the most recently used block in the internal trace table. You can specify a value in the range 1 through 262144.

If this parameter is not specified, all blocks are formatted.

You can use this parameter to reduce the storage and time requirements associated with formatting a large internal trace table in a system dump. To decide how many blocks to print to see the trace entries that you require, allow for approximately 25 entries per block.

This parameter does not apply to trace in transaction dumps, auxiliary trace, or GTF trace.

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 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=*nnnnnn-nnnnnn*).

TCB_ID=(*aaaaa*[,*aaaaa*,.,.,.])
Specifies the TCB identifiers of one or more TCBs for which trace entries are to be printed.

TCB_ADDR=(*hhhhhhhh*[,*hhhhhhhh*,.,.,.])
Specifies the address of one or more TCBs for which trace entries are to be printed.

TERMID=(*tttt*[,*tttt*,.,.,.])
Specifies the terminal identifiers 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 DFHTU710 program selects all the trace entries that are associated with any transaction-attach trace entries it finds that contain the terminal identifiers 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 59.

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 DFHTU710 program converts the store-clock (STCK) values in the trace entries to whole seconds before 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 DFHTU710 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.

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 DFHTU710 program selects all the trace entries that are associated with any transaction-attach trace entries it finds that contain the transaction identifiers 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 59.

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:

Code	Component name
AP	Application domain
BA	Business application manager
BF*	Built-in function
BM*	Basic mapping support
BR*	3270 bridge
CP*	Common programming interface
DC*	Dump compatibility layer
DD	Directory manager domain
DH	Document handling domain
DI*	Data interchange
DM	Domain manager domain
DP	Debugging profiles domain
DS	Dispatcher domain
DU	Dump domain
EC*	Event capture and emission
EI*	Exec interface

Code	Component name
EJ	Enterprise Java™ domain
EM	Event manager domain
EP	Event processing domain
FC*	File control
GC	Global catalog domain
IC*	Interval control
IE	ECI over TCP/IP domain
IS*	ISC or IRC
KC*	Task control
KE	Kernel
LC	Local catalog domain
LD	Loader domain
LG	Log manager domain
LM	Lock domain
ME	Message domain
ML	Markup language domain
MN	Monitoring domain
MP	Managed platform domain
NQ	Enqueue domain
OT	Object transaction domain
PA	Parameter domain
PC*	Program control
PG	Program manager domain
PI	Pipeline domain
PT	Partner domain
RA	Resource manager adapters
RI*	Resource manager interface (RMI)
RL	Resource life-cycle domain
RM	Recovery manager domain
RS	Region status domain
RX	RRS-coordinated EXCI domain
RZ	Request streams domain
SC*	Storage control
SH	Scheduler services domain
SJ	JVM domain
SM	Storage manager domain
SO	Sockets domain
ST	Statistics domain
SZ*	Front End Programming Interface
TC*	Terminal control
TD*	Transient data

Code	Component name
TI	Timer domain
TR	Trace domain
TS	Temporary storage domain
UE*	User exit interface
US	User domain
WB	Web domain
WU™	CICS Management Client Interface (CMCI) domain
W2	Web 2.0 domain
XM	Transaction manager domain
XS	Security manager domain

Notes:

1. Components marked * are subcomponents of the AP domain. The trace entries for these components are produced with a trace point ID of AP *nnnn*.
2. For the DS domain function CHANGE_MODE, a trace entry is generated if DS level 2 or 3 tracing is active.

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.

About this task

The association of transaction-attach trace entries with task entries is illustrated in the diagram shown in Figure 18 on page 61.

If you select trace entries by specifying the TRANID or TERMID parameters, the DFHTU710 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 18 on page 61, you can obtain formatted trace output for task IDs 00123 and 00124 by specifying the TRANID parameters. This is possible because the associated transaction-attach trace entries are present (see record numbers 2 and 7 in the diagram). Note that as the transactions in this example were task-attached, not terminal-attached, they do not have a TERMID. If they did have a TERMID, you could specify this to select the entries. 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:

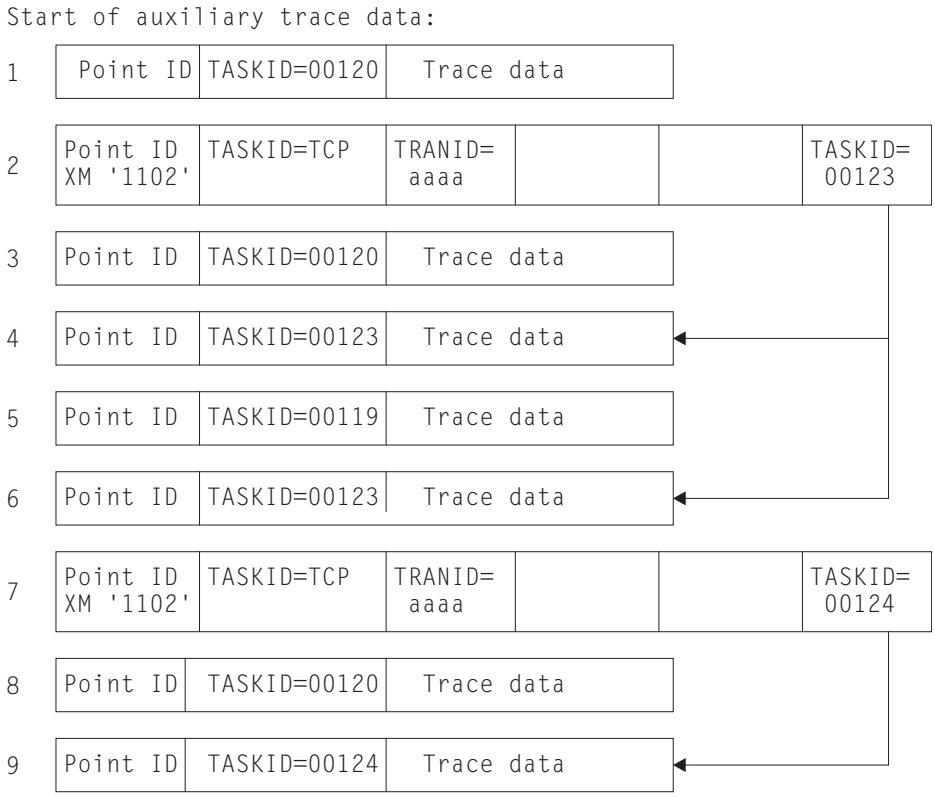
```
TERMID=LP1,                 [Select entries for terminal LP1
TRANID=(ABRW,AORD,MYTR),    [Select entries for tranids ABRW, AORD, & MYTR
TIMERG=(123000-150000))     [Select entries timed between 1230 and 1500
```

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

```
//       EXEC  PGM=DFHTU710,PARM=('TERMID=LP1,TRANID=(ABRW,AORD,MYTR)',
//         'TIMERG=(123000-150000)')
```

The following example, with the transaction ID MYTR placed in a second record, 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
```



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 18. 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.

About this task

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.

Procedure

1. You can print CICS trace entries written to GTF by invoking IPCS with the **GTFTRACE** subcommand and specifying one of the following options:
   - Specify the **USR** parameter with the event trace identifier of the records you want IPCS to select for formatting.

- Specify DFHTU710 selective trace control statements on the CICS(text) parameter.

The CICS-supplied formatting routines are called DFHTG710 and DFHTR710, supplied in CICSTS54.CICS.SDFHLINK. DFHTG710 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.

2. If several CICS regions at different CICS releases are writing to the same GTF data set, you can print the GTF trace entries using the same GTFTRACE command. To do so, you must make available the formatting routine for each CICS release. For example, for Version 5 Release 4, you could specify DFHTG710, its alias AMDUSREF, and DFHTR710.

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 DFHTR710 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, DFHTR710. You can code any of the DFHTU710 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 19 on page 63. For further information about the PROFILE PAGESIZE subcommand, see the *z/OS 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, z/OS Communications Server), see the *z/OS 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 *z/OS MVS IPCS Commands* .

Sample batch job to print CICS GTF trace entries

You can print CICS GTF trace entries by specifying the GTFTRACE subcommand of IPCS in TSO, or in a batch job as shown in this example.

```
//IPCSGTF  JOB ( accounting info),
//             CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=userid
//PRINTTR  EXEC PGM=IKJEFT01,REGION=4096K
//STEPLIB  DD DSN=CICSTS54.CICS.SDFHLINK,DISP=SHR            1
//         DD DSN=CICSTSnn.CICS.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 19. Sample IPCS job to print CICS trace entries from a GTF data set

Note:

1. The batch job in Figure 19 includes a STEPLIB statement for the load libraries that contain the modules (DFHTG710, its alias AMDUSREF, DFHTR710, DFHTG710, its alias AMDUSREF and DFHTR710) to be used to format the GTF trace entries. It also includes CICS TSnn.CICS.SDFHLINK where nn denotes an earlier release of CICS if you require to format GTF trace entries from earlier releases.
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 25 on page 79.

Invoking the CICS formatting routine from TSO

When you invoke the CICS formatting routine from TSO, the correct CICS libraries must be included in the MVS linklist and you must ensure that there is sufficient TSO storage for GTF trace formatting.

About this task

To enable the CICS formatting routines to be invoked from TSO:

Procedure

1. Copy the modules DFHTG710, AMDUSREF, DFHTR710, and DFHTRVR3 to a suitable library in the MVS linklist. Alternatively, ensure that the CICS libraries listed above are included in the MVS linklist.
2. Ensure that you have sufficient TSO storage (specified for your TSO userid when you logon) for GTF trace formatting. 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 5. Dump utilities (DFHDU710 and DFHPD710)

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:

- The transaction dump utility program, DFHDU710, for processing CICS transaction dump, as described in this section.
- The dump exit, DFHPD710, that is invoked from 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. 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: Overview" on page 70.

  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 "The SDUMP options needed to support the CICS dump exit" on page 71.)

For information about the CICS dump component keywords, and the levels that you can specify for each of them, see Summary of system dump formatting keywords and levels.

Selecting output for the transaction dump utility program (DFHDU710)

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, DFHDU710, prepares the transaction dump output for printing and prints the formatted information.

About this task

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 could 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 could be followed by further transaction dumps from the previous run.

© Copyright IBM Corp. 1989, 2017

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.

Procedure

1. Before printing a transaction dump data set, close the data set. Use the **CEMT SET DUMPDS SWITCH** command, or the corresponding EXEC CICS command, to switch to another dump data set, and close the active dump data set. To make CICS perform the switch automatically when the dump data set is full, use the **CEMT SET DUMPDS AUTOSWITCH** command, or the corresponding EXEC CICS SET command. Do not switch a dump data set while printing its contents.
2. To select the dumps that you want printed, use control statements in the SYSIN data set. In addition to the dumps you select, the DFHDU710 program always writes a summary in the form of an index of the dumps that are on the data set being processed. The index information is taken from transaction dump header records, and is written either to a data set defined with a DD name of DFHTINDX, or to the SYSPRINT data set if the DFHTINDX DD statement is missing.

Format of the SYSIN control statements

```
SELECT TYPE={OR|NOTOR|AND|NOTAND|SCAN}
[TRANID=({value|generic-value}[,value|generic-value}],.,.)]
[DUMPCODE=({value|generic-value}[,{value|generic-value}],.,.)]
[DUMPID=({value|value-range}[,{value|value-range}],.,.)]
[PAGESIZE=(value)]
[TIME=({time|time-range}[,{time|time-range}],.,.)]
[UPPERCASE=YES]
END
```

Figure 20. SYSIN control statements for the DFHDU710 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 can 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 DFHDU710 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 DFHDU710 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 for the transaction dump utility program (DFHDU710)

The job stream to run the DFHDU710 program should include DD statements for the following data sets:

DFHDMPDS (mandatory)
The input data set, from which the dump data is to be processed.

DFHPRINT (mandatory)
The output data set, usually a printer, to which the dump data is written.

DFHTINDX (optional)
The output data set to which the dump index summary is written, needed if you want the index summary output. If you omit the DFHTINDX DD statement, DFHDU710 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 DFHDU710 program, needed if you want to code specific selection parameters. If you omit the SYSIN DD statement, you get message IEC130I.

The following example is a sample job stream for the DFHDU710 program.

```
//PRNTDMP  JOB  accounting info,name,MSGLEVEL=(1,1),
//              CLASS=A,MSGCLASS=A,REGION=2M                          1
//         EXEC PGM=DFHDU710,PARM='command,command,...'  2
//STEPLIB  DD   DSN=CICSTS54.CICS.SDFHLOAD,DISP=SHR   3
//DFHDMPDS DD   DSN=CICSTS54.CICS.DFHDMPA,DISP=SHR
//DFHTINDX DD   SYSOUT=A
//DFHPRINT DD   SYSOUT=A,DCB=(BLKSIZE=133)
//SYSPRINT DD   SYSOUT=A
//SYSIN    DD   *
 SELECT TYPE=OR
 [selection parameters, each on a separate line]
 END
/*
//
```

Figure 21. Sample job to format and print CICS transaction dump data sets

1. The sample JCL gives a region size of 2 MB that you might typically need to run the DFHDU710 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 DFHDU710 utility in your CICS environment.
2. 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 DFHTU710 utility program for details).

   Note: ABBREV and FULL are not valid keywords of theDFHDU710 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.
3. 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.

Using IPCS to format and analyze CICS dumps: Overview

The interactive problem control system (IPCS) provides MVS installations with an interactive facility for diagnosing software failures.

About this task

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 73.

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

Preparing to use IPCS to format CICS SDUMPs

About this task

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" on page 72.)
- Ensure that the CICS-supplied dump exit routines can be found by your IPCS job. (See "Making the required CICS dump exit routines available" on page 72.)

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
All prefixed storage area for all processors

COUPLE
Cross-system coupling facility (XCF) information

CSA Common service area

GRSQ Global Resource Serialization control blocks

LPA Link pack area modules

NUC Non-page-protected areas of the DAT-on nucleus

RGN Private area of the address space being dumped (this includes the LSQA and SWA)

SQA System queue area

SUMDUMP
Summary dump

TRT GTF, system trace, and master trace data

WLM Workload Manager areas

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.

About this task

The IPCS default table, BLSCECT, usually in the SYS1.PARMLIB library, has the following entry for CICS:

```
IMBED MEMBER(DFHIPCSP)  ENVIRONMENT(ALL) /* CICS                */
```

Procedure

1. 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. For example:

   ```
   //IPCSPARM  DD  DSN=SYS1.PARMLIB,DISP=SHR         For BLSCECT
   //          DD  DSN=CICSTS54.CICS.SDFHPARM,DISP=SHR     For DFHIPCSP
   ```

2. You must ensure that the names of the IPCS exit routines, specified by the EP(name) operands in the DFHIPCSP member, match the names of the CICS-supplied release-specific IPCS exit routines.

Example

The CICS-supplied DFHIPCSP member, installed in the CICSTS54.CICS.SDFHPARM library, contains the CICS release-specific entries for the IPCS exit control table. These entries are listed in the following code example.

```
/* ================================================================ */
    EXIT EP(DFHPD660) VERB(CICS660) ABSTRACT(+
   'CICS Transaction Server for z/OS V4 R1 analysis')
  EXIT EP(DFHPD670) VERB(CICS670) ABSTRACT(+
   'CICS Transaction Server for z/OS V4 R2 analysis')
  EXIT EP(DFHPD680) VERB(CICS680) ABSTRACT(+
   'CICS Transaction Server for z/OS V5 R1 analysis')
  EXIT EP(DFHPD690) VERB(CICS690) ABSTRACT(+
   'CICS Transaction Server for z/OS V5 R2 analysis')
  EXIT EP(DFHPD700) VERB(CICS700) ABSTRACT(+
   'CICS Transaction Server for z/OS V5 R3 analysis')
  EXIT EP(DFHPD710) VERB(CICS710) ABSTRACT(+
   'CICS Transaction Server for z/OS V5 R4 analysis')
/* ================================================================ */
```

Figure 22. Release-specific entries in DFHIPCSP for DFHPDnnn routines

Making the required CICS dump exit routines available

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.

About this task

The dump exit routine for CICS TS 5.4, DFHPD710 , is installed in the SYS1.CICSTS54.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, DFHPD710.

Procedure

Add the library containing the dump exit routine to the MVS linklist. This ensures that your IPCS job can find the appropriate dump exit routine to format the CICS system dump data.

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.

About this task

You can use the **TRS** to select trace entries by:

- A specified number of blocks at the end of the internal trace table
- Kernel task
- Task identifier
- Terminal
- Transaction identifier
- Time period
- Trace identifier

For more information about the statements that you can use to select parts of the CICS internal trace table, see "The trace selection parameters for DFHTU710" on page 54.

For abbreviated and full trace formatting, a trace summary table provides summary information about the trace entries that relate to each task in the system during the time period. The trace summary table appears at the end of the formatted trace output. Use the table to see the tasks that were traced, and the location and number of their trace entries in the trace output. The table also highlights any long time gaps between the trace entries for a task, which can indicate a performance problem, and any exception trace entries for a task. The trace summary table is not produced for short-format trace.

Procedure

1. To select the parts of the internal trace to be formatted by IPCS, specify the **TRS** parameter on the IPCS VERBEXIT command. For example:

```
VERBEXIT CICS710 'DEF=1,DLI=1,KE=3,TR=2,TRS=<TRANID=CSSC,KE_NUM=12,LAST_BLOCKS=500>'
```

   The VERBEXIT statement specifies the verb name CICS710 to process CICS Transaction Server for z/OS, Version 5 Release 4 system dump data. This corresponds to the IPCS dump exit routine DFHPD710, as specified in the DFHIPCSP member in the CICSTS54.CICS.SDFHPARM library.

2. Specify the **TR** parameter, with a value other than 0, on the IPCS VERBEXIT command. This setting ensures that output from the trace domain is formatted and the **TRS** keyword is effective.

Using CICS-supplied dump exit routines to format CICS SDUMPs

To use IPCS to format a CICS SDUMP you must specify the CICS dump exit parameters on the **VERBEXIT** subcommand of IPCS.

About this task

Procedure

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 "Specifying DFHIPCSP CICS exit control data needed" on page 72 for the CICS-supplied verb names.
The syntax of the CICS exit parameters is shown in Figure 23 and described in "The CICS dump exit parameters."

Example

For some examples of using IPCS to process CICS SDUMPs, see "Examples of the use of level numbers" on page 76.

```
[JOB={jobname|CURRENT}]
[UPPERCASE]

[,DEF={0|1|2|3}]
[,keyword [=levelnumber]]
```

Figure 23. The CICS dump exit parameters

where

keyword
specifies the CICS component ID. .

levelnumber
specifies the level of data to be output, either to a terminal or to a printer.

The CICS dump exit parameters

The descriptions of 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 CICS710 '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 CICS710 '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

There are two parts of a CICS system dump that are not governed by component keywords, and are therefore outside the scope of the DEF parameter.

These 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 CICS710 '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 "The dump summary and error index" on page 77.

Examples of the use of level numbers

VERBEXIT CICS710 'JOB=CURRENT,KE=1,DS' prints a summary of the kernel domain data, and all available information for the dispatcher domain.

VERBEXIT CICS710 '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 CICS710 without any parameters produces summary and control block output for all the CICS components in the dump.

The CICS710 dump exit component keywords

The component keywords specify which functional areas of the CICS dump you want the CICS710 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 DFHTU710 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 DFHTU710 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 Summary of system dump formatting keywords and levels.

The CICS710 dump exit can be used in either a batch job or interactively. For an example of a batch IPCS job, see Figure 26 on page 80. For information about using IPCS, see the following MVS IPCS manuals:

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

Trace entry selection

You can select trace entries from internal trace in a system dump for formatting and printing by using the TRS component keyword. You do this in a similar way to the selection of trace entries in an auxiliary trace for formatting and printing.

For the TRS keyword to be effective, the TR keyword value must be 1, 2, or 3.

The trace selection parameters can be any valid trace selection parameters available to DFHTU710 for the formatting of CICS auxiliary trace entries, except for the parameters **PAGESIZE**, **ABBREV**, **SHORT**, and **FULL**. You can also use the **LAST_BLOCKS** parameter, which is specifically for formatting internal trace in a system dump. As with DFHTU710, you can select any number of parameters from those available.

The trace selection parameters have the same format and default values when used to select trace entries from an internal SDUMP trace, as when you use DFHTU710 to format auxiliary trace entries. You must use angled brackets around the parameter, or sequence of parameters, that you specify. For example:

```
VERBEXIT DFHPD710 'TR=2,TRS=<LAST_BLOCKS=100,TASKID=00123'>
```

For descriptions of available parameters, see "The trace selection parameters for DFHTU710" on page 54.

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 24 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 24. 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

You can use the following two sample jobs to process CICS SDUMPs using IPCS.

The first sample, in Figure 25 on page 79, is an example of how to create an IPCS dump directory; the second sample, in Figure 26 on page 80, 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 CICS710 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 data set for
//*       use as an IPCS dump directory.  There are two job steps:
//*
//*       1. CREDDIR creates a dump directory data set for use by IPCS
//*
//*       2. INITDIR initializes the data set.
//*************************************************************
//CREDDIR   EXEC PGM=IDCAMS
//SYSPRINT  DD   SYSOUT=*
//AMSDUMP   DD   SYSOUT=*
//SYSIN     DD   *
 DEFINE CLUSTER (NAME(CICSTS54.CICS.IPCSDIR)             -          1
                 VOLUMES(volid)                    -                2
                 CYLINDERS(2 1)                    -
                 BUFFERSPACE(65536)                -
                 KEYS(128 0))                      -
                 DATA( NAME(CICSTS54.CICS.IPCSDIR.DATA)   -         1
                 CONTROLINTERVALSIZE(4096))        -
                 INDEX(NAME(CICSTS54.CICS.IPCSDIR.INDEX))           1
/*
//INITDIR   EXEC PGM=IKJEFT01,REGION=0M
//SYSTSPRT  DD SYSOUT=*
//SYSTSIN   DD *
IPCSDDIR 'CICSTS54.CICS.IPCSDIR'                                    1
END
/*
//
```

Figure 25. Sample job to create an IPCS dump directory

Note:

1. Change 'CICSTS54.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 26 on page 80 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=0M
//STEPLIB   DD DSN=CICSTS54.CICS.SDFHLINK,DISP=SHR                    1
//          DD DSN=CICSTS54.CICS.SDFHPARM,DISP=SHR
//DFHSDUMP  DD DSN=DUMP.NAME,DISP=SHR                                 2
//DFHSNAP   DD SYSOUT=*                                               3
//IPCSDDIR  DD DSN=CICSTS54.CICS.IPCSDIR,DISP=SHR                     4
//IPCSPARM  DD  DSN=SYS1.PARMLIB,DISP=SHR                             5
//          DD  DSN=CICSTS54.CICS.SDFHPARM,DISP=SHR
//IPCSTOC   DD SYSOUT=*                                               6
//IPCSPRNT  DD SYSOUT=*
//SYSPROC   DD DSN=SYS1.SBLSCLI0,DISP=SHR                             7
//SYSTSPRT  DD SYSOUT=*
//IPCSDUMP.SYSTSIN   DD *
IPCS NOPARM
DROPDUMP DD(DFHSDUMP)                                               8
SETDEF DD(DFHSDUMP) NOPROBLEM NOCONFIRM NOTERMINAL PRINT LIST
VERBEXIT CICS710 'UPPERCASE,DEF=1,DLI=2,KE=3,TR=2'                  9
END
/*
```

Figure 26. Sample job to format a CICS SDUMP using IPCS and the CICS dump exit

Note:

1. The batch job in Figure 26 includes STEPLIB statements for:
   - The CICSTS54.CICS.SDFHLINK library, to enable the VERBEXIT subcommand of IPCS to invoke the CICS-supplied IPCS dump exit routine DFHPD710.
   - The CICSTS54.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 'CICSTS54.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 Starting up CICS regions in Administering.
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 CICS710 to process CICS TS for z/OS, Version 5.4 system dump data. This corresponds to the IPCS dump exit routine DFHPD710, as specified in the DFHIPCSP member in the CICSTS54.CICS.SDFHPARM library.

Chapter 6. Monitoring dictionary utility program (DFHMNDUP)

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 performance 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 which you can modify or adapt to your own purposes — it is intended to show you how you can code your own monitoring utility program to print CICS monitoring data. See Chapter 7, "Sample monitoring data print program (DFH$MOLS)," on page 89.

This section describes the purpose of the DFHMNDUP program, with a sample job to run it.

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.

Generating a performance dictionary record using DFHMNDUP

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.

Before you begin

If you require programming information about the performance dictionary and data records before performing this task, see Performance class data. For information about the MCT definitions, see Monitoring control table (MCT).

© Copyright IBM Corp. 1989, 2017

About this task

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.

Procedure

1. 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 sets you are processing and which contain performance data.
2. Specify the appropriate parameters in the SYSIN data set to control the DFHMNDUP program, so that it can generate the correct dictionary record for the performance data you are processing. The values that you specify 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 4.

Table 4. 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 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 a description of these parameters, see "Parameters for the DFHMNDUP program" on page 85.

Example

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
/*
```

Parameters for the DFHMNDUP program

You can use the following parameters to control the information produced by the DFHMNDUP program.

DATE=yyddd or DATE=yyyyddd
Specifies the Julian date to be included in the dictionary record, where:

yy represents the year of the twenty-first century (for example 16 for 2016).

yyyy represents the year.

ddd represents the day, in the range 1 through 366.

For example 17355 represents the 20th December 2016 and the date 1996354 represents the 20th December 1996. If you do not specify a date, the current date is used.

GAPPLID=name
Specifies the APPLID specified on either the **APPLID** system initialization parameter, or the generic APPLID in an XRF environment for which you are analyzing performance data.

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 twenty-first century (for example, 16 for 2016).

yyyy represents the year.

ddd represents the day, in the range 1 through 366.

For example 16355 represents the 20th December 2016 and the date 1996354 represents the 20th December 1996. If you do not specify a date, the current date is used.

JOBNAME=xxxxxxxx
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 performance 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 z/OS Communications Server 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 *z/OS MVS Installation Exits*.

Sample job illustrating the use of DFHMNDUP

The following sample job 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=CICSTS54.CICS.SDFHLOAD,DISP=SHR                  1
//        DD DSN=mct.table.loadlib,DISP=SHR
//SYSUT4  DD DSN=CICSTS54.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=2003205
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.SMFDMPnn,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=CICSTS54.CICS.SDFHLOAD,DISP=SHR
//INPUT    DD DSN=CICSTS54.CICS.applid.MNDUPREC,DISP=OLD          5
//         DD DSN=SYS1.SMFDMPnn,DISP=OLD
//SYSPRINT DD SYSOUT=A
//SYSABEND DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//SYSOUT   DD SYSOUT=A
//SORTDIAG DD SYSOUT=A
//SYSIN    DD *
/*
```

Figure 27. Sample job stream to run DFHMNDUP

(DFH$MOLS control statements - see "DFH$MOLS control statements summary" on page 92)

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 could 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 *z/OS 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.

Chapter 7. Sample monitoring data print program (DFH$MOLS)

When you have unloaded monitoring data from SMF data sets for one or more CICS regions, you can print the data using the DFH$MOLS sample utility program, or a utility program of your own that is modeled on the DFH$MOLS sample program.

CICS provides two programs for processing any CICS monitoring data that is written to system management facilities (SMF) data sets:

- **DFHMNDUP**: A utility program that generates a performance dictionary record, in a sequential data set, for use with monitoring data extracted from SMF data sets. See Monitoring dictionary utility program (DFHMNDUP).
- **DFH$MOLS**: A print program for CICS monitoring data. DFH$MOLS is a sample program that you can modify or adapt to your own purposes. It shows you how you can code your own monitoring utility program to print CICS monitoring data.

The DFH$MOLS program is a data reduction program that produces reports from the data collected by the CICS monitoring domain (MN) and written to SMF data sets.

DFH$MOLS can identify any SMF 110 monitoring records that have been compressed and expand them using the z/OS Data Compression and Expansion Services (CSRCESRV), before working with them. If you specify the EXPAND control statement, DFH$MOLS copies the compressed monitoring records to an output data set in their expanded format, with the records that were not compressed. Other reporting tools can use the output data set of SMF 110 monitoring records.

The default is for monitoring records to be compressed, COMPRESS=YES.

You can switch data compression off and on in a number of ways. In the Monitoring Control Table (MCT), specify the COMPRESS option in the DFHMCT TYPE=INITIAL macro. Use the monitoring facility transaction CEMN or the CEMT or EXEC CICS SET MONITOR command to change the data compression option.

A monitoring record with a compressed data section is identified by the compressed record length in the SMFMNCRL field in the SMF product section, which is present only for a compressed record.

In CICS Transaction Server for z/OS, Version 5 Release 3, DFH$MOLS can process SMF 110 monitoring data records for the following supported releases:

- CICS Transaction Server for z/OS, Version 5 Release 3
- CICS Transaction Server for z/OS, Version 5 Release 2
- CICS Transaction Server for z/OS, Version 5 Release 1
- CICS Transaction Server for z/OS, Version 4 Release 2
- CICS Transaction Server for z/OS, Version 4 Release 1

However, the UNLOAD control statement (which unloads performance class monitoring data into a fixed length record format) can be used only with

© Copyright IBM Corp. 1989, 2017

monitoring data for CICS Transaction Server for z/OS, Version 3 Release 2 onwards. Any version or release of DFH$MOLS cannot process monitoring data for a version or release *later* than itself, so you must always use the DFH$MOLS from the highest version or release available to you.

For programming information about the structure of CICS SMF type 110, and how the monitoring data is packaged in the SMF records, see CICS monitoring facility: Performance and tuning.

The DFH$MOLS program requires a performance dictionary record to process monitoring performance data. When it locates a dictionary record, it builds an in store dictionary and processes the subsequent (if any) performance data using this dictionary. Whenever it reads a new dictionary record, the current dictionary is released and a new in store dictionary is built. The dictionary record must come before any related performance data; otherwise, the DFH$MOLS program abends. Monitoring exception records do not require a dictionary, so they can precede the first dictionary record and still be successfully processed. For more information about performance dictionary records, see Generating a performance dictionary record using DFHMNDUP.

Processing CICS monitoring data

The job steps involved in processing CICS monitoring data are:

Procedure

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 *z/OS MVS System Management Facilities (SMF)* manual.
2. Run the DFH$MOLS program to print monitoring records, which you can optionally select by means of control statements. DFH$MOLS automatically sorts the monitoring data for you. 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:

   a. A dictionary data section, consisting of a sequence of dictionary entries.
   b. An exception data section, consisting of a single exception record.
   c. A performance data section, consisting of a sequence of field connectors followed by one or more performance records.
   d. A transaction resource data section, consisting of one or more transaction resource monitoring records.
3. Optional: 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. 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 “DFH$MOLS control statements summary” on page 92.

Example

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.

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 *z/OS DFSORT Application Programming Guide*.

Sample job stream for DFH$MOLS

Figure 28 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')              1
//OUTDD1   DD DSN=&&TEMP,DISP=(NEW,PASS),SPACE=(CYL,(2,1)),           2
//            UNIT=SYSDA
//SYSPRINT DD SYSOUT=A
//SYSIN    DD *
         INDD(INDD1,OPTIONS(DUMP))
         OUTDD(OUTDD1,TYPE(110(1)))                                  3
/*
//*
//PRNT     EXEC PGM=DFH$MOLS
//STEPLIB  DD DSN=CICSTS54.CICS.SDFHLOAD,DISP=SHR                     4
//INPUT    DD DSN=&&TEMP,DISP=(OLD,DELETE),UNIT=SYSDA
//SYSUT2   DD DSN=&&XPND,DISP=(NEW,PASS),UNIT=SYSDA,                  5
//            SPACE=(CYL,(25,10))
//SYSUT4   DD DSN=&&UNLD,DISP=(NEW,PASS),UNIT=SYSDA,                  6
//            SPACE=(CYL,(25,10))
//SORTWK01 DD SPACE=(CYL,(5,1)),UNIT=SYSDA                            7
//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 *
:
Control statements for data selection                               8
:
OPTION LOCAL                                                        9
*
PRINT ALL                                                           10
RESOURCE ALL                                                        11
*
EXPAND                                                              12
*
UNLOAD LOCAL                                                        13
/*
//
```

Figure 28. Sample job to unload and process CICS data from SMF data sets

Note:

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 in "Job to run the DFHSTUP program" on page 33.

2. If you want to keep the unloaded data set, change the DSN and DISP parameters appropriately.
3. The SMF data set can contain any type of SMF record, but in this example, only CICS type 110 records are unloaded. Although these records might include CICS statistics records, and any CICS user journal records written to SMF, the DFH$MOLS program ignores those and processes monitoring data only; these monitoring records are identified by a record subtype of 01.
4. If you generated your own version of the DFH$MOLS program and stored it in a different library from the CICS-supplied version, change the STEPLIB statement accordingly. On the STEPLIB statement, specify the library that contains an appropriate version of the DFH$MOLS program for the CICS release of SMF records to be formatted.Chapter 7, "Sample monitoring data print program (DFH$MOLS)," on page 89 lists the earlier CICS Transaction Server versions and releases supported by the CICS TS for z/OS, Version 5.4 release of DFH$MOLS. Always use the DFH$MOLS from the highest version or release available to you.
5. Defines the output data set for the EXPAND control statement.
6. Defines the output data set for the UNLOAD control statement.
7. These sort work files are required for sorting SMF 110 monitoring data records.
8. Specify the control statements for data selection and other options in SYSIN. For details of the control statements for the DFH$MOLS program, see "DFH$MOLS control statements summary."
9. OPTION LOCAL is recommended so that the monitoring record start and stop timestamp fields are converted into local time in the reports produced.
10. PRINT ALL specifies that you want to print all types of monitoring SMF type 110 records that are selected by other control statement options.
11. RESOURCE ALL specifies that you want to print all types of resource data within each resource record.
12. EXPAND specifies that you want the monitoring data to be written to an output data set, including any compressed SMF 110 monitoring records in their expanded format. The DFH$MOLS program processes only CICS SMF 110 records with record subtype 01. All other SMF 110 records are ignored.
13. UNLOAD LOCAL specifies that the performance class monitoring data is to be unloaded into a fixed length record format.

DFH$MOLS control statements summary

You can control the processing that the DFH$MOLS program performs by specifying control statements.

For full details about these statements, see "DFH$MOLS control statement descriptions" on page 94 and "Rules for coding DFH$MOLS control statements" on page 94.

***** [user comments]

BREAK
Use this statement to group SELECT and 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

If yy is coded, the twenty-first century is assumed; for example, 16 for 2016.

EXPAND

DDNAME=xxxxxxxx[,NEWDCB]

Use this option if some or all of the input monitoring records were compressed, and you want to create an output data set with these records in their expanded format, with the records that were never compressed. If you want to print reports only or to unload the records into a fixed-length format, do not specify the EXPAND option. DFH$MOLS identifies and expands any compressed monitoring records automatically before working with them. Specify only the EXPAND option if you want to create an output data set of SMF 110 monitoring records.

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 with SELECT statements to form SELECT and IGNORE groups. (See the BREAK control statement.)

OPTION GMT|LOCAL

DFH$MOLS prints the monitoring record start and stop time-stamp fields in GMT, or converts them into local time.

PRINT

ALL, DIC, EXC, IDN, PER, RES

RESOURCE

ALL, DPL, FILE, TSQUEUE

Use this option to enable selective printing of the resource type data in the transaction resource monitoring records.

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 with IGNORE statements to form SELECT and IGNORE groups. See the BREAK control statement for more information.

TIME

START=hh.mm.ss,STOP=hh.mm.ss

TIMEOFF

This option is accepted for upgrade purposes, but it is now ignored. The

option was used to specify the suppression of testing for data being out of sequence, but now DFH$MOLS always sorts the input monitoring data.

UNLOAD{DDNAME=xxxxxxxx[,LOCAL]}
Use this statement to unload the input performance class monitoring data into the fixed-length record format. The UNLOAD control statement can be used with monitoring data for CICS Transaction Server for z/OS, Version 5 Release 4 only, and not with monitoring data for any earlier CICS releases.

Rules for coding DFH$MOLS 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.

DFH$MOLS control statement descriptions

The rules when processing control statements, and the order they must be applied in, are listed. The control statements that you can specify to control the DFH$MOLS program processing are described in detail.

Rules for processing control 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$MOLS program processes the SELECT/IGNORE groups in the order in which you specify them in SYSIN.
3. APPLID, USERID, TERMID, TRANID, PRCSTYPE, and TASKNO parameters can all be included in the same run.
4. You cannot specify SELECT and IGNORE for the same parameter in the same SELECT/IGNORE group. For example, specifying SELECT TERMID and IGNORE TERMID is invalid.
5. If you specify SELECT for more than one parameter in a SELECT/IGNORE group, the SELECT statements form a logical AND function.
6. If you specify IGNORE for more than one parameter in a SELECT/IGNORE group, the IGNORE statements form a logical OR function.
7. The DFH$MOLS program processes all SELECT statements in a group before any IGNORE statements in the same group.

8. If a data record satisfies all 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 the SELECT/IGNORE statements in a group are 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 are 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.

Control statements

*** (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 ends the previous (ungrouped) 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 purpose 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, so 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.

CONTROL STOPAFT=*nnnnnnnn*
Specifies the number of records that you want to process. The **STOPAFT** parameter limits the number of SMF type 110 records that you want the DFH$MOLS program to process. The DFH$MOLS program ends after processing the number of SMF 110 records specified by *nnnnnnnn*.

DATE
Specifies the start and stop dates, which, with the TIME statement (if specified), enable 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.

Note:

1. CICS dictionary records are always processed by the DFH$MOLS program and are not affected by any date and 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.

EXPAND
Specifies that the monitoring data is written to an output data set, including any compressed SMF 110 monitoring records in their expanded format, with the monitoring records that were never compressed. For more information about record types processed by DFH$MOLS, see note 12 in Sample job stream for DFH$MOLS. The output data set of SMF 110 monitoring records can be used by reporting tools that cannot use the z/OS Data Compression and Expansion Services (CSRCESRV) to expand compressed records.

A monitoring record with a compressed data section is identified by the compressed record length in the SMFMNCRL field in the SMF product section, which is present only for a compressed record.

If you want to just print reports, or to unload the monitoring records into a fixed-length format, you do not need to specify the EXPAND option. DFH$MOLS identifies and expands any compressed monitoring records automatically before working with them. You only need to specify the EXPAND option when you want to create an output data set of SMF 110 monitoring records.

DDNAME=*name*
Specifies the ddname for the output data set to hold the SMF 110 monitoring records. If you do not code this keyword, the default ddname SYSUT2 is used, and your job stream must include a SYSUT2 DD statement. If you code this keyword to specify a different ddname, your job stream must include the corresponding DD statement.

NEWDCB
To ignore the DCB information from the original data set, specify NEWDCB. Supply the new DCB information on the JCL for the output data set.

Note:

1. When the EXPAND control statement is specified, the only parameter for IGNORE and SELECT statements that operates during creation of the output data set is the APPLID option. The PRCSTYPE, TASKNO, TERMID, TRANID, and USERID parameters are ignored while the output data set is being produced. You can also select monitoring records for the output data set by date, using the DATE parameter, or by time, using the TIME parameter.

2. Monitoring data is not automatically printed when the EXPAND control statement is specified. If this statement is specified, and you also want to print monitoring data, you must specify the PRINT control statement explicitly. When you specify the PRINT statement to print monitoring records, all the selection parameters on your IGNORE and SELECT statements now operate for the selection of the monitoring records for printing.

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=*xxxxxxxx*[*,yyyyyyyy,..*.]

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

PRCSTYPE=*xxxxxxxx*[*,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=*xxxx*[*,yyyy,..*.]

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

TRANID=*xxxx*[*,yyyy,..*.]

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

USERID=*xxxxxxxx*[*,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=.

If you are using the EXPAND statement, the only parameter for IGNORE and SELECT statements that operates during creation of the output data set is the APPLID option. The PRCSTYPE, TASKNO, TERMID, TRANID, and USERID parameters are ignored while the output data set is being produced. If you specify the PRINT statement to print monitoring records, all the selection parameters on your IGNORE and SELECT statements now operate for the selection of the monitoring records for printing.

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 time stamp fields in GMT time in the reports produced.

LOCAL

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

PRINT {ALL|DIC|EXC|IDN|PER|RES} or PRINT {*rectype1*,*rectype2*,...,*rectypen*}
Specifies the type of monitoring data record that you want to print in full.

If you do not specify the PRINT statement, the default action is that all the monitoring SMF type 110 records that are selected by other control statement options are printed in full. The exception is when the UNLOAD or EXPAND control statements are specified. In these cases, monitoring data is not automatically printed. If these statements are specified, and you also want to print monitoring data, you specify the PRINT control statement explicitly.

If you want to print more than one type of record in full, but not all, you can specify them as a list separated by commas, as shown in the second form of the PRINT parameter. In this way, you can specify any combination, such as:

```
PRINT DIC,PER
PRINT EXC,PER
PRINT DIC,PER,RES
PRINT IDN,RES,EXC
PRINT RES,PER,EXC
```

When you specify a list of record types in this way, the records that you select are printed in full. For the record types that you omit, the SMF headers and SMF product sections are printed for records that are selected by other control statement options, but the data sections of the records are not printed.

ALL List all the monitoring SMF type 110 records that are selected by other control statement options. This is the default if you omit the PRINT statement.

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

EXC List only the monitoring exception class records that are selected by other control statement options.

IDN List only the identity class records that are selected by other control statement options.

PER List only the monitoring performance class records that are selected by other control statement options.

RES List only the transaction resource monitoring records that are selected by other control statement options.

RESOURCE {ALL|DPL|FILE|TSQUEUE}
Specifies the type of resource data that you want to print in each resource record.

ALL Print all types of resource data. ALL is the default if you omit the RESOURCE statement.

DPL Print only the distributed program link type of resource data.

FILE Print only the file type of resource data.

TSQUEUE
Print only the temporary storage type of resource data.

Note: The identification resource data is printed regardless of which option is selected.

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=*xxxxxxxx*[,*yyyyyyyy*,.,.]
Specify one or more generic APPLIDs to include monitoring data from the CICS regions identified by these APPLIDs.

PRCSTYPE=*xxxxxxxx*[,*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=*xxxx*[,*yyyy*,.,.]
Specify one or more terminal identifiers to include monitoring data associated with these terminals.

TRANID=*xxxx*[,*yyyy*,.,.]
Specify one or more transaction identifiers to include monitoring data for these transactions.

USERID=*xxxxxxxx*[,*yyyyyyyy*,.,.]
Specify one or more user identifiers to include monitoring data for transactions submitted by these users.

TIME
Specifies the start and stop times, which, with the DATE statement (if specified), enable 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. Therefore, the program might select performance records for times that are a few minutes outside the specified period because of the way that 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 `hhmmss`. A start time is optional, and, if omitted, the report includes all records for the start date, regardless of time.

STOP=*stop-time*
The end time of the period for which you want records processed, in the form `hh.mm.ss` or `hhmmss`. An end time is optional, and, if omitted, the report includes all records for the stop-date, regardless 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.

UNLOAD {DDNAME=*xxxxxxxx*[,LOCAL]}
Specifies that the performance class monitoring data is unloaded into a fixed-length record format. The format of the output data set can be mapped using the copy member DFHMNPDA provided in CICSTS54.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 converts the performance class start and stop time stamp fields into local time in the output performance class records.

Note:

1. If the UNLOAD control statement is specified, the SORT control statement must also be specified.
2. Monitoring data is not automatically printed when the UNLOAD control statement is specified. If this statement is specified, and you also want to print monitoring data, specify the PRINT control statement explicitly.
3. The UNLOAD control statement can be used only with monitoring data for CICS Transaction Server for z/OS, Version 5 Release 4 , and not with monitoring data for any earlier CICS releases.

Control statement 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.

The DFH$MOLS program pads, with trailing blanks, operands that have fewer 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.

The following control statements exclude all records for transaction CEMT, regardless of user ID, and exclude all records containing userid OP7 (regardless of transaction ID). All other records are included.

```
IGNORE TRANID=CEMT
IGNORE USERID=OP7
```

The following control statements include only records for transaction CEMT, except for those from terminal TRM3.

```
SELECT TRANID=CEMT
IGNORE TERMID=TRM3
```

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.

If you are using the EXPAND statement, the only parameter for IGNORE and SELECT statements that operates during creation of the output data set is the APPLID option. The PRCSTYPE, TASKNO, TERMID, TRANID, and USERID parameters are ignored while the output data set is being produced. If you specify the PRINT statement to print monitoring records, all the selection parameters on your IGNORE and SELECT statements now operate for the selection of the monitoring records for printing.

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.

The following control statements include only records with a transaction identifier of CEMT *and* with a terminal identifier of TRM3. Any other records are not included.

```
SELECT TERMID=TRM3
SELECT TRANID=CEMT
```

The following control statements include only those records that are from the CICS region with the generic APPLID DBDCCICS, and that are for transaction CEMT, but that do *not* have the terminal identifier TRM3.

```
SELECT APPLID=DBDCCICS
SELECT TRANID=CEMT
IGNORE TERMID=TRM3
```

Abend codes and error messages for DFH$MOLS

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; RC='nn'; 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 number of the return code from DFSORT is given in the message (RC='nn'). The SORT utility writes a message to SYSOUT defining the error.

For information about these error messages, see the *z/OS DFSORT Messages, Codes and Diagnosis Guide*.

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, transaction resource 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 Chapter 6, "Monitoring dictionary utility program (DFHMNDUP)," on page 83 and Monitoring dictionary record.)

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.

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 28 on page 91.

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 28 on page 91.

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

114: UNABLE TO OPEN DDNAME ‘xxxxxxxx’; REPORT IS TERMINATED

Explanation

The DFH$MOLS program was unable to open the data set specified on the DD statement used for the UNLOAD control statement. ‘xxxxxxxx’ is either SYSUT4, the default, or the ddname specified by the DDNAME= parameter on the UNLOAD control statement. Ensure that the JCL for the job was correct. A sample set of JCL to execute the DFH$MOLS sample program is contained in Figure 28 on page 91.

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

116: NO FILE RESOURCE ENTRIES

Explanation

The DFH$MOLS program did not find any file resource entries in the CICS monitoring records.

117: NO TSQUEUE RESOURCE ENTRIES

Explanation

The DFH$MOLS program did not find any temporary storage queue resource entries in the CICS monitoring records.

118: UNABLE TO EXPAND A COMPRESSED RECORD, RC='nn'; REPORT IS TERMINATED

Explanation

The DFH$MOLS program was unable to expand the compressed data section for an SMF 110 monitoring record. This abend is issued when the z/OS Data Compression and Expansion service CSRCESRV FUNCTION=EXPAND was unable to expand the data section in the SMF record. For more information on the return codes issued by the CSRCESRV service, see the *z/OS MVS Assembler Services Reference ABE-HSP*.

This message is followed by an MVS abend U118 with a dump.

119: UNABLE TO OPEN DDNAME '*xxxxxxxx*'; REPORT IS TERMINATED

Explanation

The DFH$MOLS program was unable to open the data set specified on the DD statement used for the EXPAND control statement. '*xxxxxxxx*' is either SYSUT2, which is the default, or the ddname specified by the DDNAME= parameter on the EXPAND 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 28 on page 91.

This message is followed by an MVS abend U119 without a dump.

120: UNEXPECTED CSRCESRV QUERY ERROR, RC='nn'; REPORT IS TERMINATED

Explanation

The DFH$MOLS program received an unexpected (non-zero) return code from the z/OS Data Compression and Expansion service CSRCESRV FUNCTION=QUERY. For more information on the return codes issued by the CSRCESRV service, see the *z/OS MVS Assembler Services Reference ABE-HSP*.

This message is followed by an MVS abend U120 with a dump.

121: NO DPL RESOURCE ENTRIES

Explanation

The DFH$MOLS program did not find any distributed program link request resource entries in the CICS monitoring records.

Chapter 8. Load module scanner (DFHEISUP)

The load module scanner utility can scan load libraries for the CICS commands in load modules, and identify which modules contain specific API or SPI commands.

For example, if you know that a particular API or SPI command must be changed to take advantage of a new feature in CICS, you can use the load module scanner to identify all the load modules that contain the command. The load module scanner can look for specific commands, or options on commands, or combinations of options. It can also scan for commands where certain options are not specified.

The load module scanner is a batch utility. It locates all the EXEC CICS commands in your load modules, and then applies a filter that you create, to identify only the commands in which you are interested. It returns one of two types of report:

- A summary report, giving a list of the modules that contain the commands that are specified by your filter, and the number of specified commands in each module.
- A detailed report, with a list for each module to show which of the specified commands it contains, at which offsets, including EDF information if available.

Both types of report identify the language of each load module.

CICS supplies a sample job, DFHEILMS in SDFHSAMP, that you can edit and use to execute the load module scanner. On any run of the load module scanner, you can scan all the load modules in a load library (if you specify a complete PDS), or you can specify certain load modules within the library to scan. A useful technique is to run a summary scan for a particular load library to produce a list of the modules that contain the commands in which you are interested. You can place this list of modules into a data set. You can then run a detailed scan, using this data set as input, to produce a detailed report on only the modules that contain the commands in which you are interested.

You specify the commands to be reported upon as a filter input file. You can name the commands in which you are interested. You can also include or exclude commands that have certain parameters. The 'any' operator can be used to search for all parameters, or for all commands. A number of sample command lists are provided to help you. Use DFHEIDBR to search for commands that are not supported in the 3270 bridge environment. Use DFHEIDTH to search for commands that give access to shared storage, which could make a program not threadsafe. Use DFHEIDAL to search for all the basic valid commands, and use DFHEIDNT to find all non-threadsafe CICS commands (which would cause a switch to the QR TCB).

Points to bear in mind when you use the load module scanner

- When you use the load module scanner, as with all types of scanners, it is not possible to guarantee that a hit is genuine. There might be a few false hits, because of static data in a module. The reports identify candidates that match a set of criteria (the commands and options that are specified in your filter), and the candidates that are identified might not be an exact match to your requirements.
- The load module scanner is designed to scan up to about 100 load modules on a single run. The sample job, DFHEILMS in SDFHSAMP, specifies a region size of

© Copyright IBM Corp. 1989, 2017

512M, which is appropriate for this amount of scanning. If you need to scan more load modules on a single run, you can increase the region size. However, do not specify REGION=0M on the job to run the load module scanner, as this can lead to performance problems. If you have many load modules in a load library that you want to scan, you might prefer to copy smaller batches of the load modules into new load libraries, and scan the new load libraries.

- To ensure sufficient storage below the 16MB line, specify the Language Environment run time option ALL31(ON). ALL31(ON) is the default Language Environment run time option for CICS.
- The PL/I compiler V3.6 changed the default compiler option CEESTART from CEESTART(FIRST) to CEESTART(LAST). For CEESTART(LAST), an ENTRY CEESTART card must be included when linking the MAIN routine as described in the PL/I compiler documentation. The ENTRY CEESTART card is used by DFHEISUP to find API or SPI commands. Alternatively, you can specify CEESTART(FIRST) explicitly.

Figure 29 shows how the scanner works.

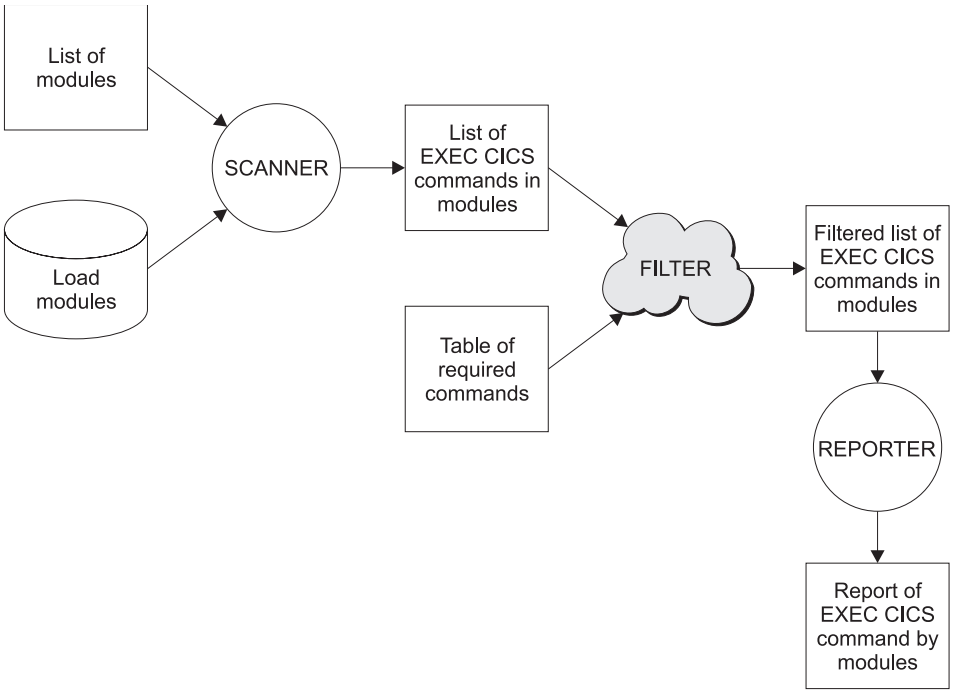


Figure 29. Load module scanner with filter

Creating a filter input file for the load module scanner

About this task

The load module scanner, DFHEISUP, locates all EXEC CICS commands in the load modules in the library you are scanning. It then applies a filter that you have specified, and reports only on the subset of commands that you have listed in the filter. For example, you might want to find a particular command that is not supported for certain usage in CICS. You can specify that command in a filter input file. When you run a summary report using that filter, the load module scanner reports the names of the modules containing that command. When you

run a detailed report using that filter, the load module scanner reports each occurrence of that command in each module.

Each command that you want to find is placed on a separate line in the filter input file, for example:

- RETURN *
- WRITEQ TS *

Note that where a command consists of two words, you must specify both words. For example, just specifying INQUIRE * on a line in a filter input file does not work, because INQUIRE is not valid on its own as a command. You need to specify the complete command, for example, INQUIRE FILE *. A sample filter input file, DFHEIDAL, is supplied to provide a definitive list of all valid CICS commands that the load module scanner will recognize.

As well as the command itself, you can specify parameters or options that you want to find or exclude. When specifying parameters, two operators are allowed, as follows:

- not interested in commands containing this parameter

* any parameter

If no parameter or operator is specified for a particular command, the load module scanner searches for that command with no parameters. In the example above, RETURN * finds all RETURN commands, with any parameters or no parameters. RETURN, without the *, would only find occurrences of the RETURN command with no parameters.

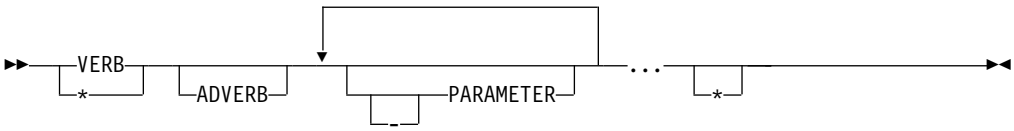


Table 5 shows you how commands can be specified for the filter input file.

Table 5. Examples of commands for the filter input file

Command	Parameter specification
SYNCPOINT	SYNCPOINT commands that have no parameters
RETURN *	All RETURN commands, with any parameters or no parameters
WRITEQ TS *	All WRITEQ TS commands, with any parameters or no parameters
LINK PROGRAM SYSID *	LINK PROGRAM commands with SYSID, with any other parameters (or no other parameters)
GETMAIN BELOW -SHARED *	GETMAIN commands with BELOW, but without SHARED, with any other parameters (or no other parameters)
GETMAIN -INITIMG *	GETMAIN commands without INITIMG, with any other parameters (or no other parameters)

Table 5. Examples of commands for the filter input file (continued)

Command	Parameter specification
WRITEQ TS QUEUE FROM LENGTH	WRITEQ TS commands that have only the parameters QUEUE & FROM & LENGTH (and no other parameters)
* LENGTH *	All commands with a LENGTH parameter and possibly other parameters
* LENGTH	All commands with a LENGTH parameter and no other parameters
* *	All commands with any parameters
*	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, unless you want the line to imply all commands with only a LENGTH parameter.

If you include * * or * as a line in the filter input file, this effectively means that no filtering of commands takes place.

Comments can be placed in the filter input file by beginning the line with a # character.

Each line specified in the file will be checked to ensure it represents a valid filter. Checks are made to ensure that 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 you incorrectly specify a command or parameter, you receive an error message, and the load module scanner terminates with a return code of 12.

CICS provides the following sample tables in the SDFHSAMP library:

- DFHEIDBR contains the filter set for commands that are not supported by the 3270 bridge.
- DFHEIDTH contains the filter set for commands that give access to shared storage. The use of these commands could make a program not threadsafe, unless it has the necessary synchronization logic in place to ensure serialization and prevent concurrent update.
- DFHEIDAL contains the filter set for all possible commands.
- DFHEIDNT contains the filter set for CICS commands that are not threadsafe and will cause a switch to the QR TCB. This could influence performance when trying to run in the Open Transaction Environment.

Note: The following exceptions apply to the load module scanner:

Misdetection of verbs

SPOOLOPEN INPUT is detected as SPOOLOPEN OUTPUT, and WAIT JOURNAL is detected as WAIT JOURNALNUM. SPOOLOPEN OUTPUT and WAIT JOURNALNUM, where present, are detected as themselves.

INQUIRE UOWENQ is 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 is 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 are 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. The reports identify candidates matching a given set of criteria (the commands and options specified in your filter), and the candidates that are identified might not be an exact match to your requirements.

Creating a summary report using the load module scanner

You can request a summary report from the load module scanner by editing and running the job DFHEILMS in SDFHSAMP. The summary report gives a list of the modules that contain the commands specified by your filter, and the number of specified commands in each module.

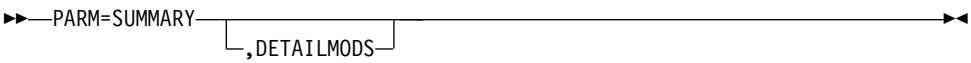
About this task

The job to request a summary report includes statements identifying the filter to be used (DFHFLTR), and (optionally) a data set to be produced for use with a detailed report (DFHDTL).

To request a summary report from the scanner:

Procedure

1. Ensure that the load library to be scanned does not contain concatenated data sets.
2. Edit the JOB accounting parameters as appropriate.
3. Ensure that your region size is appropriate for the number of load modules that you want to scan. Do not specify REGION=0M. See Points to bear in mind when using the load module scanner for more information.
4. On the PARM statement, specify SUMMARY. If you want to produce a list of modules to be used when running a detailed scan on this load library, specify DETAILMODS as well.



SUMMARY
Specifies that a summary scan (and report) is required for the entire library, except for certain CICS modules (those beginning with XDFH, and the DL/I modules ASMTDLI, CBLTDLI and PLITDLI), CICS tables, and those modules that cannot be loaded (due to an error).

DETAILMODS
Specifies that the names of those modules that may contain at least one of the EXEC CICS commands identified in your filter, are to be written to the sequential file defined by the DFHDTL DD statement.

5. Edit the STEPLIB, DFHIN, SYSPRINT, DFHFLTR and DFHDTL statements as required.

STEPLIB DD statement
Specifies the name of the load library where you have installed the scanner program, DFHEISUP.

DFHIN DD statement
Specifies the name of the load library to be scanned. This must not contain concatenated data sets. The DFHIN DD statement accepts both PDS and PDSEs.

SYSPRINT DD statement
Specifies the destination for the summary report.

DFHFLTR DD statement
Specifies the input file for the filter, containing details of the commands in which you are interested. CICS provides the following sample tables in the SDFHSAMP library:

- DFHEIDBR contains the filter set for commands that are not supported by the 3270 bridge.
- DFHEIDTH contains the filter set for commands that give access to shared storage. The use of these commands could make a program not threadsafe, unless it has the necessary synchronization logic in place to ensure serialization and prevent concurrent update.
- DFHEIDAL contains the filter set for all possible commands.
- DFHEIDNT contains the filter set for CICS commands that are not threadsafe and will cause a switch to the QR TCB. This could influence performance when trying to run in the Open Transaction Environment.

See "Creating a filter input file for the load module scanner" on page 108 for information on how to create your own filter input file.

DFHDTL DD statement
This optional statement specifies the name of a sequential data set where the list of modules with potential command matches is to be sent. To send the list of modules to the data set, specify DETAILMODS on the PARM statement. You can edit the data set to alter the list of modules to be scanned, but note that all the modules listed must be in the same load library. You can use this file to restrict a subsequent detailed report on that load library, 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 using the load module scanner" on page 114.

Example

Figure 30 on page 113 shows an example of a job to run a summary scan, using the sample command list DFHEIDBR as the filter input file.

```
//DFHSCNR  JOB (accounting information)
//DFHSCAN    EXEC PGM=DFHEISUP,PARM=('SUMMARY, DETAILMODS'),REGION=512M
//STEPLIB  DD DSN=HLQ.SDFHLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSERR   DD SYSOUT=*
//* Filter file, supplied by user, or from samples data set
//DFHFLTR  DD DSN=HLQ.ADFHSAMP(DFHEIDBR),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.SDFHLOAD,DISP=SHR
```

Figure 30. Example of a job to run a summary scan

Figure 31 shows an example of a summary report.

```
CICS LOAD MODULE SCANNER UTILITY
SCAN PERFORMED ON Tue Jun  6 13:37:48 2000 USING TABLE RSTABLE1.3

SUMMARY LISTING OF UTL.CL717.LOAD

Module Name  Commands Found  Language
DFHLMS00                15  Assembler
INTROC                   8  COBOL

LOAD LIBRARY STATISTICS

Total modules in library                              =      2
Total modules Scanned                                 =      2
Total CICS modules/tables not scanned                 =      0
Total modules possibly containing requested commands =      2
```

Figure 31. Example of a summary report produced by the scanner

Each summary report contains the following:

- A line for each module in the library, containing:
  - Module name. If a module contains more than 1024 commands, the name of the module is included several times, once for every 1024 commands found, until all commands have been found.
  - The number of possible specified commands located.
  - The identified language of the module.

    Note: For both detailed and summary reports, the identified language of the module is based on the language of the first identified command. If the module is a mixed language module, only the first located language is reported. In addition, if the first identified command is a false hit, the language for that module will be reported incorrectly.
- The total count of:
  - Modules in the library
  - Modules scanned
  - CICS modules and tables (not scanned)
  - Modules that could contain some specified commands

Creating a detailed report using the load module scanner

You can request a detailed report from the scanner by editing and running the job DFHEILMS in SDFHSAMP. The detailed report gives a list for each module to show which of the specified commands it contains, at which offsets, including EDF information if available.

About this task

The job to request a detailed report includes statements identifying a filter to be used (DFHFLTR), and (optionally) a data set containing a list of relevant modules to be scanned (DFHLIST).

Procedure

1. Ensure that the load library to be scanned does not contain concatenated data sets.
2. Edit the JOB accounting parameters as appropriate.
3. Ensure that your region size is appropriate for the number of load modules that you want to scan. Do not specify REGION=0M. See Points to bear in mind when using the load module scanner for more information.
4. On the PARM statement, specify DETAIL. If you want to scan all modules in the load library, specify ALL as well. If you want to restrict the detailed scan by using a data set containing a list of modules identified during a summary scan of the load library, do not specify ALL.



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 the requested EXEC CICS commands. If ALL is omitted, only those modules listed in the data set specified on the DFHLIST DD statement are to be scanned.

5. Edit the STEPLIB, DFHIN, SYSPRINT, DFHFLTR and DFHLIST statements as required. You can restrict the modules to be scanned during a detail scan, by using a data set containing a list of modules identified during a summary scan. If you want to do this, specify the name of the data set on the DFHLIST statement, and also ensure that the load library named in DFHIN is the same as the load library named when the summary scan was run.

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 data sets. The DFHIN DD statement accepts both PDS and PDSEs. If you use the DFHLIST DD statement to name a data set created during a summary scan, ensure that DFHIN is set to the same location as when the summary scan was run. The scanner looks for the modules listed in DD DFHLIST in the location specified by DD DFHIN.

SYSPRINT DD statement
Specifies the destination for the detailed report.

DFHFLTR DD statement
Specifies the input file for the filter, containing details of the commands in which you are interested. CICS provides the following sample tables in the SDFHSAMP library:

- DFHEIDBR contains the filter set for commands that are not supported by the 3270 bridge.
- DFHEIDTH contains the filter set for commands that give access to shared storage. The use of these commands could make a program not threadsafe, unless it has the necessary synchronization logic in place to ensure serialization and prevent concurrent update.
- DFHEIDAL contains the filter set for all possible commands.
- DFHEIDNT contains the filter set for CICS commands that are not threadsafe and will cause a switch to the QR TCB. This could influence performance when trying to run in the Open Transaction Environment.

See "Creating a filter input file for the load module scanner" on page 108 for information on how to create your own filter input file.

DFHLIST DD statement
Specifies the name of an optional data set containing a list of the modules to be scanned. The modules in the list are those that potentially include the commands you specified in your filter. The data set can be created during a summary run of the scanner using the same filter, and can be edited — see "Creating a summary report using the load module scanner" on page 111 for the procedure to create the data set. The file contains the fully qualified data set name and the member name of each module to be scanned. Each module name must appear on a separate line. All the modules must be within the load library that you are scanning (the PDS specified in the DFHIN DD statement).

If you want to perform a detail scan on all the modules in the load library, specify ALL on the PARM statement, and change the DFHDTL DD statement to specify //DFHLIST DD DUMMY.

Example

Figure 32 shows an example of a job to run a detailed scan, using the sample command list DFHEIDBR as the filter input file.

```
//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 data set
//DFHFLTR  DD DSN=HLQ.ADFHSAMP(DFHEIDBR),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 32. Example of a job to run a detailed scan

Figure 33 on page 116 shows the contents of a detailed report.

```
  CICS LOAD MODULE SCANNER UTILITY
SCAN PERFORMED ON Tue Jun  6 08:47:51 2000 USING TABLE RSTABLE1.3

DETAILED LISTING OF UTL.CL717.LOAD

Module Name        DFHLMS00
Module Language    Assembler
Offset/EDF         Command
-----------------  ----------------------------------------
00000648/no-edf    ISSUE ERROR STATE CONVID
00000668/no-edf    LOAD PROGRAM
00000677/no-edf    LOAD PROGRAM HOLD
00000686/no-edf    LOAD PROGRAM LENGTH
00000695/no-edf    LOAD PROGRAM FLENGTH
00000704/no-edf    LOAD PROGRAM LENGTH HOLD
00000713/no-edf    LOAD PROGRAM FLENGTH HOLD
00000722/no-edf    WRITEQ TS FROM LENGTH QUEUE AUXILIARY NUMITEMS
00000731/no-edf    WRITEQ TS FROM LENGTH QUEUE AUXILIARY
00000740/no-edf    WRITEQ TS FROM LENGTH QUEUE MAIN
00000749/no-edf    SEND TEXT FROM LENGTH TERMINAL
00000764/no-edf    SEND TEXT FROM LENGTH L40 TERMINAL
00000779/no-edf    SEND TEXT FROM LENGTH L64 TERMINAL
00000794/no-edf    SEND TEXT FROM LENGTH L80 TERMINAL
00000809/no-edf    RETURN

Module Name        INTROC
Module Language    COBOL
Offset/EDF         Command
-----------------  ----------------------------------------
00000174/00085     ADDRESS COMMAREA
00000204/00164     LINK PROGRAM COMMAREA LENGTH
00000221/00132     RETURN
00000238/00127     RETURN
00000255/00112     RETURN
00000272/00106     RETURN
00000289/00100     RETURN
00000306/00080     RETURN

Total possible commands located = 23

LOAD LIBRARY STATISTICS

Total modules in library                              =      2
Total modules Scanned                                 =      2
Total CICS modules/tables not scanned                 =      0
Total modules possibly containing requested commands =      2
```

Figure 33. Example of a detailed report produced by the scanner

Each detailed report contains a section for each module, with the following details:

- A header part for each module, giving its name and language.

  Note: For both detailed and summary reports, the identified language of the module is based on the language of the first identified command. 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.
- A line for each specified command located, giving:
  - The offset of the command's Arg0 declaration from the start of the load module. (This offset is in decimal format.) Typically, a command's Arg0 declaration is in a different part of the load module from the actual command, so the offset does not indicate where the command is located.

– If available, the EDF line number at which the command was located, otherwise “no-EDF” is returned.
– What the command appears to be, for example, **DELETEQ TS QUEUE**. This information includes identified specified parameters.

In a program written in Assembler, identical commands using identical parameters share the same Arg0 in the literal pool. DFHEISUP only reports one of these identical commands.

- The total count of the number of possible commands located.
- Library totals, as with the summary report, but only for those modules selected for the detailed run.

Activating trace for the load module scanner

Trace should be enabled only at the request of your IBM® service representative.

About this task

To activate trace:

Procedure

Prefix the PARMs in the JCL with the required trace type. One of the following supported trace types can be specified — TI, TD, or TB. The following example shows a summary run including the trace type TB.

```
//DFHSCNR  JOB (accounting information)
//DFHSCAN  EXEC PGM=DFHEISUP,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 data set
//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.SDFHLOAD,DISP=SHR
```

Figure 34. Example of a summary scan modified to activate trace

Chapter 9. Batch-enabling sample programs for RLS access-mode data sets (DFH0BATx)

If you have recoverable VSAM data sets that are open in RLS access mode to CICS regions and you want to update them from a batch application, you must typically quiesce the data sets before using them in batch mode. This section describes some procedures that you can use to help you to automate the process of preparation for batch processing.

Before you begin

If you do not want to quiesce your recoverable VSAM data sets, you can use DFSMStvs which allows online updating of VSAM recoverable data sets. Using DFSMtvs allows CICS full read and write access to any VSAM data sets that it is using. For more information about DFSMStvs, see the z/OS manual, DFHSMStvs Planning and Operating or the IBM Redbooks® publication DFSMStvs Overview and Planning Guide.

About this task

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.

The DFH0BATx sample programs described in this section are provided by CICS to 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.

For more information about the procedures you should follow for checking and handling retained locks when switching to non-RLS mode, see Resolving retained locks on recoverable resources.

© Copyright IBM Corp. 1989, 2017

Preparing data sets for batch operations using the DFH0BATx sample programs

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.

About this task

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 indoubt 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 6. 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 in COBOL, 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 7. Functional summary of the DFH0BATx programs

Program	Functional overview
DFH0BAT1	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. 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.
DFH0BAT2	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. 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. This process of issuing DPL requests to DFH0BAT7 and DFH0BAT5 is repeated for each of the data set names obtained from BATD.
DFH0BAT3	DFH0BAT3 is invoked by transaction BAT3 in the CICS region chosen to be the coordinator region. It initiates the forced backout of any indoubt 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. For each data set, DFH0BAT3 issues a DPL request to DFH0BAT6, in each target CICS region, to force the backout of indoubt 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. This process of issuing DPL requests to DFH0BAT6 and DFH0BAT8 is repeated for each of the data set names obtained from BATD.
DFH0BAT4	Linked by DPL request from DFH0BAT1 to disable specified transactions.

Table 7. Functional summary of the DFH0BATx programs (continued)

Program	Functional overview
DFH0BAT5	Linked by DPL request from DFH0BAT2 to gather and return retained lock information to its caller.
DFH0BAT6	Linked by DPL request from DFH0BAT3 to force the backout of indoubt units of work.
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 DFH0BATx 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**.

About this task

Preparing input for the DFH0BATx sample programs

The three coordinating programs require input from extrapartition transient data queues. These TD queues provide the parameters the sample programs need.

About this task

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

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.

To prepare these TD queues and the control information:

Procedure

1. Define the sequential data sets for the TD queues as fixed block data sets with an 80 byte block size. You can define these in the CSD.
   - 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.

2. 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.

Chapter 10. 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.

About this task

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 could be:

- CICS-supplied DFH macros listed in the .
- CICS-supplied macros present in MACLIB.
- User-modified CICS macros.
- User-written macros.
- None of these.

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 (see "How DFHMSCAN works").

DFHMSCAN identifies CICS DFH macros explicitly where it can. It also reports the use of obsolete **EXEC CICS ADDRESS CSA** commands.

The primary purpose of DFHMSCAN 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.

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.

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.

© Copyright IBM Corp. 1989, 2017

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.

JCL to run DFHMSCAN

To run DFHMSCAN, use the following JCL:

```
//SCANJOB  JOB ACCOUNTING INFO,CLASS=A
//SCAN     EXEC PGM=DFHMSCAN,PARM='pppppppp'
//STEPLIB  DD DSN=CICSTS54.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 "Producing a summary report from DFHMSCAN."

'NAME1,NAME2,...'
The DFHMSCAN program scans the named modules and produces a detailed report for each one. See "Producing a detailed report from DFHMSCAN" on page 127.

DSN=xxxxxxx.LOADLIB
"xxxxxxx" is the load module library to be scanned. Only one load module library can be specified.

Producing a summary report from DFHMSCAN

If you specify `PARM='$SUMMARY'`, DFHMSCAN summarizes the entire library.

Results

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 ("Total no. modules scanned")
  - Macro-level programs ("Number of macro programs")

- Macro-level programs of each type (“Assembler macro programs”, “COBOL macro programs”, and “PL/I macro programs”)
- Programs that possibly contain EXEC CICS ADDRESS CSA commands (“Address CSA programs”).

Example

Figure 35 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 35. Example of a summary report produced by DFHMSCAN

Producing a detailed report from DFHMSCAN

If you specify `PARM='NAME1,NAME2,...'`, DFHMSCAN scans the named modules only, and produces a detailed report for each module and a summary report.

About this task

Results

- 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 11. 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.

For information about the end-of-day time and the system initialization parameter to set it, see "Setting the statistics recording status, end-of-day time and recording interval" on page 31.

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$STED. To change them, you must modify the source of DFH$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 started COLD, 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.

© Copyright IBM Corp. 1989, 2017

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 values for parameters of DFH$STED

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 12. 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.

Monitoring statistics are lost if an immediate shutdown is implemented. This is a result of the shutdown, and any data in the monitoring buffer is lost.

If the program named by the shutdown transaction cannot be loaded, CICS waits indefinitely for all user tasks to complete, which might 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, DFHCESD

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. z/OS Communications Server is force closed, IRC and TCP/IP are 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,

© Copyright IBM Corp. 1989, 2017

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)           Number of tasks in the system at last sample
```

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 data set 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 z/OS Communications Server persistent sessions support is being used (that is, unless the persistent session delay interval is set to a value greater than zero), z/OS Communications Server is force closed, IRC and TCP/IP are 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 shutdown assist programs

Source code versions of the default program are provided in Assembler, COBOL, and PL/I, in the CICSTS54.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 the restart manager (ARM) is active, and you use the default shutdown assist program during shutdown processing, ARM restarts the CICS region after the region has completed shutting down. If you do not want ARM to restart the CICS region, amend the sample shutdown assist program by adding the NORESTART option to the command EXEC CICS PERFORM SHUT IMMEDIATE.

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.

Considerations for your shutdown assist program

The shutdown assist task is a system task that behaves differently from a user task. If all user tasks have ended, shutdown continues to completion as quickly as possible, and CICS does not wait for the shutdown assist program to complete.

If your shutdown assist program loses control and then all user tasks terminate, control might not return to the program before shutdown completes. For example, if an EXEC CICS DELAY, WAIT EXTERNAL or SUSPEND command is coded in the program, and all user tasks complete before the interval elapse time expires, then the program does not regain control. You should bear this in mind if recoverable resources are updated by your program, because if the program loses control and then all user tasks complete, full commit and backout protection cannot be guaranteed. The EXEC CICS commands which might cause the program to lose control are not limited to those mentioned here.

Specifying REQID(DFHCESD) on an EXEC CICS DELAY command coded in your program results in early expiration of the unexpired DELAY request if all user tasks complete during the delay interval. In this case, the program regains control from the EXEC CICS command even though all non-system tasks have finished. However, bear in mind that the full interval elapse time might not have expired when the program regains control. If the EXEC CICS DELAY command is issued after all user tasks have completed, it might never expire, because CICS shutdown continues without resuming the system task.

Figure 36 on page 134 shows some example messages generated by a run of the DFHCESD sample program.

```
16.14.08 JOB09120  +DFHTM1715 IYLX1 CICS is being quiesced by userid CICSUSER in transaction CEMT
                              at netname IGCS21F.
16.14.08 JOB09120  +DFHDM0102I IYLX1 CICS is quiescing.
16.14.08 JOB09120  +DFHTM1781 IYLX1 CICS shutdown cannot complete because some non-system user tasks
                              have not terminated.
16.14.09 JOB09120  +DFHCESD  IYLX1    SHUTDOWN ASSIST TRANSACTION CESD STARTING. SHUTDOWN IS NORMAL.
16.14.09 JOB09120  +DFHCESD  IYLX1    LIST OF SHUNTED UNITS OF WORK IN THE SYSTEM FOLLOWS.
16.14.09 JOB09120  +DFHCESD  IYLX1    SHUNTED TRANSACTION RFI4, TERMID S21F, UNIT OF WORK ABE7194B52539603
16.16.09 JOB09120  +DFHCESD  IYLX1    THERE ARE NOW 0100 TASKS STILL IN THE SYSTEM.
16.16.26 JOB09120  +DFHDU0303I IYLX1 Transaction Dump Data set DFHDMPA closed.
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID KK07, TERMID     , USERID CICSUSER, TASKNO 000113
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID KK15, TERMID     , USERID CICSUSER, TASKNO 000119
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID KK16, TERMID     , USERID CICSUSER, TASKNO 000120
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID KK23, TERMID     , USERID CICSUSER, TASKNO 000127
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID KK24, TERMID     , USERID CICSUSER, TASKNO 000128
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID KK29, TERMID     , USERID CICSUSER, TASKNO 000133
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID KK36, TERMID     , USERID CICSUSER, TASKNO 000139
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID KK38, TERMID     , USERID CICSUSER, TASKNO 000140
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID KK47, TERMID     , USERID CICSUSER, TASKNO 000144
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID SK07, TERMID     , USERID CICSUSER, TASKNO 000161
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID SK15, TERMID     , USERID CICSUSER, TASKNO 000167
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID SK16, TERMID     , USERID CICSUSER, TASKNO 000168
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID SK23, TERMID     , USERID CICSUSER, TASKNO 000175
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID SK24, TERMID     , USERID CICSUSER, TASKNO 000176
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID SK29, TERMID     , USERID CICSUSER, TASKNO 000181
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID SK36, TERMID     , USERID CICSUSER, TASKNO 000187
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID SK38, TERMID     , USERID CICSUSER, TASKNO 000188
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID SK44, TERMID     , USERID CICSUSER, TASKNO 000189
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID SK47, TERMID     , USERID CICSUSER, TASKNO 000192
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID SK49, TERMID     , USERID CICSUSER, TASKNO 000194
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RSD1, TERMID S234, USERID CICSUSER, TASKNO 000418
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID SK49, TERMID     , USERID CICSUSER, TASKNO 000424
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID SK47, TERMID     , USERID CICSUSER, TASKNO 000426
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID SK44, TERMID     , USERID CICSUSER, TASKNO 000429
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID SK23, TERMID     , USERID CICSUSER, TASKNO 000437
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID SK16, TERMID     , USERID CICSUSER, TASKNO 000444
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X134, USERID CICSUSER, TASKNO 000531
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X150, USERID CICSUSER, TASKNO 000532
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X141, USERID CICSUSER, TASKNO 000533
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X124, USERID CICSUSER, TASKNO 000534
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X147, USERID CICSUSER, TASKNO 000535
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X152, USERID CICSUSER, TASKNO 000536
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X178, USERID CICSUSER, TASKNO 000537
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X158, USERID CICSUSER, TASKNO 000538
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X168, USERID CICSUSER, TASKNO 000540
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X127, USERID CICSUSER, TASKNO 000542
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X176, USERID CICSUSER, TASKNO 000543
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X122, USERID CICSUSER, TASKNO 000545
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X149, USERID CICSUSER, TASKNO 000546
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X145, USERID CICSUSER, TASKNO 000547
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X117, USERID CICSUSER, TASKNO 000548
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X128, USERID CICSUSER, TASKNO 000550
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X126, USERID CICSUSER, TASKNO 000551
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X162, USERID CICSUSER, TASKNO 000552
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X143, USERID CICSUSER, TASKNO 000553
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X164, USERID CICSUSER, TASKNO 000554
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X133, USERID CICSUSER, TASKNO 000555
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X121, USERID CICSUSER, TASKNO 000556
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X113, USERID CICSUSER, TASKNO 000558
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X160, USERID CICSUSER, TASKNO 000559
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X170, USERID CICSUSER, TASKNO 000560
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X120, USERID CICSUSER, TASKNO 000561
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X115, USERID CICSUSER, TASKNO 000562
```

Figure 36. Example messages generated by the sample DFHCESD program

```
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X154, USERID CICSUSER, TASKNO 000567
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X112, USERID CICSUSER, TASKNO 000568
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X177, USERID CICSUSER, TASKNO 000569
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X156, USERID CICSUSER, TASKNO 000570
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X125, USERID CICSUSER, TASKNO 000571
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X130, USERID CICSUSER, TASKNO 000572
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X155, USERID CICSUSER, TASKNO 000573
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X119, USERID CICSUSER, TASKNO 000574
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X174, USERID CICSUSER, TASKNO 000575
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID SK51, TERMID     , USERID CICSUSER, TASKNO 000576
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X142, USERID CICSUSER, TASKNO 000577
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X179, USERID CICSUSER, TASKNO 000578
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X132, USERID CICSUSER, TASKNO 000579
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X135, USERID CICSUSER, TASKNO 000580
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X118, USERID CICSUSER, TASKNO 000581
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X175, USERID CICSUSER, TASKNO 000582
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X131, USERID CICSUSER, TASKNO 000583
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X163, USERID CICSUSER, TASKNO 000584
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X159, USERID CICSUSER, TASKNO 000585
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X111, USERID CICSUSER, TASKNO 000586
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X151, USERID CICSUSER, TASKNO 000587
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X169, USERID CICSUSER, TASKNO 000588
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X171, USERID CICSUSER, TASKNO 000589
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X166, USERID CICSUSER, TASKNO 000590
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID SK07, TERMID     , USERID CICSUSER, TASKNO 000591
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X153, USERID CICSUSER, TASKNO 000593
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X114, USERID CICSUSER, TASKNO 000594
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X148, USERID CICSUSER, TASKNO 000596
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X173, USERID CICSUSER, TASKNO 000597
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X136, USERID CICSUSER, TASKNO 000598
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X165, USERID CICSUSER, TASKNO 000599
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X116, USERID CICSUSER, TASKNO 000600
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X123, USERID CICSUSER, TASKNO 000601
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X144, USERID CICSUSER, TASKNO 000602
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X129, USERID CICSUSER, TASKNO 000603
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X172, USERID CICSUSER, TASKNO 000604
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X167, USERID CICSUSER, TASKNO 000605
16.16.26 JOB09120  +DFHCESD  IYLX1    PURGING TRANID RFS1, TERMID X138, USERID CICSUSER, TASKNO 000606
16.16.27 JOB09120  +DFHCESD  IYLX1    LIST OF SHUNTED UNITS OF WORK IN THE SYSTEM FOLLOWS.
16.16.27 JOB09120  +DFHCESD  IYLX1    SHUNTED TRANSACTION RFI4, TERMID S21F, UNIT OF WORK ABE7194B52539603
16.16.28 JOB09120  +DFHCESD  IYLX1    THERE ARE NOW 0081 TASKS STILL IN THE SYSTEM.
16.16.31 JOB09120  +DFHCESD  IYLX1    THERE ARE NOW 0073 TASKS STILL IN THE SYSTEM.
16.16.43 JOB09120  +DFHCESD  IYLX1    THERE ARE NOW 0072 TASKS STILL IN THE SYSTEM.
16.16.45 JOB09120  +DFHCESD  IYLX1    THERE ARE NOW 0070 TASKS STILL IN THE SYSTEM.
16.16.47 JOB09120  +DFHCESD  IYLX1    THERE ARE NOW 0004 TASKS STILL IN THE SYSTEM.
16.17.05 JOB09120  +DFHCESD  IYLX1    LIST OF SHUNTED UNITS OF WORK IN THE SYSTEM FOLLOWS.
16.17.05 JOB09120  +DFHCESD  IYLX1    SHUNTED TRANSACTION RFI4, TERMID S21F, UNIT OF WORK ABE7194B52539603
16.17.07 JOB09120  +DFHZC2316 IYLX1 VTAM ACB is closed
16.17.09 JOB09120  +DFHCESD  IYLX1    THERE ARE NOW 0002 TASKS STILL IN THE SYSTEM.
16.17.09 JOB09120  +DFHTM1782I IYLX1 All non-system tasks have been successfully terminated.
16.17.30 JOB09120  +DFHRM0131 IYLX1 Resynchronization required with IRC resources.
16.17.30 JOB09120  +DFHRM0131 IYLX1 Resynchronization required with LU62 resources.
16.17.30 JOB09120  +DFHRM0131 IYLX1 Resynchronization required with IND resources.
16.17.40 JOB09120  +DFHRM0203 IYLX1 There are 1 indoubt, 0 commit-failed and 0 backout-failed UOWs.
16.17.56 JOB09120  +DFHRM0130 IYLX1 Recovery manager has successfully quiesced.
16.18.01 JOB09120  +DFHKE1799 IYLX1    TERMINATION OF CICS IS COMPLETE.
```

Note: VTAM is now z/OS Communications Server.

Chapter 13. Recovery manager utility program (DFHRMUTL)

This section 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 Controlling start and restart.

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.

You can specify what you want DFHRMUTL to do by supplying parameters in a single optional record in the input data set, SYSIN. See “Specifying parameters for DFHRMUTL” on page 138.

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.

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.
- The next CICS start type found on the global catalog, DFHGCD.
- Counts of those units of work (UOWs) that were in an indoubt, commit-failed or backout-failed state when CICS was shut down. Note that these counts are provided if the next CICS start type is warm.
- 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.

© Copyright IBM Corp. 1989, 2017

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 for DFHRMUTL

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 141 contains some example jobs that illustrate the uses of DFHRMUTL.

DD statements

The DD statements for the input and output data sets used by DFHRMUTL are:

STEPLIB DD
Defines a partioned 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 for DFHRMUTL

You can use the parameters **SET_AUTO_START** and **COLD_COPY** to control the actions that DFHRMUTL takes.

About this task

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.

A CICS system that was shut down warm, and which had no indoubt, commit-failed or backout-failed units of work keypointed at that time, can be restarted cold without loss of data integrity. However, if the system was not shut down in a controlled manner, or indoubt, commit-failed or backout-failed units of work were recorded on the system log, then a cold restart should not be performed, because system data integrity could be compromised.

The DFHRMUTL summary information which is output to SYSPRINT can be used when determining whether AUTOCOLD is suitable or not. DFHRMUTL also issues the information message DFHRM0315I, "AUTOCOLD or AUTOINIT should not be used", if it detects that this condition applies.

If the input catalog is empty (or has insufficient information in it) AUTOCOLD is not allowed.

The AUTOCOLD option has no effect on the following system initialization parameters (which are taken from the local catalog), unless they are specified as SIT overrides at CICS startup:

- DSALIM
- DUMP
- EDSALIM
- STGPROT
- SUBTSKS
- TRANISO

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 Dealing with a corrupt system log.

Note:

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 AUTOINIT option has no effect on the following system initialization parameters (which are taken from the local catalog), unless they are specified as SIT overrides at CICS startup:

- DSALIM
- DUMP
- EDSALIM
- STGPROT
- TRANISO

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.

Note:

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 from DFHRMUTL

DFHRMUTL sets one of the following return codes:

00 The parameters are valid and all reads and writes to the input and output data sets were successful.

16 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

About this task

Figure 37 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.

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 Setting up the catalog data sets.

```
//RMUTL    EXEC PGM=DFHRMUTL,REGION=1M
//STEPLIB  DD DSNAME=hlq.SDFHLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=A
//DFHGCD   DD DSNAME=catalog.dataset,DISP=OLD
//SYSIN    DD *
  SET_AUTO_START=AUTOINIT
/*
```

Figure 37. DFHRMUTL—setting the next auto start to be an initial start.

Examining the override record

About this task

Figure 38 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 38. 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

About this task

Figure 39 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 39. DFHRMUTL—resetting a warm or emergency start

Improving the performance of a cold start

About this task

Figure 40 on page 143 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 40. DFHRMUTL—setting the global catalog for a cold start. COLD_COPY is used to improve performance.

Chapter 14. BMS macro generation utility program (DFHBMSUP)

This section describes the BMS macro generation utility, DFHBMSUP, to recreate BMS macro statements from a mapset load module.

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. 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.

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 for DFHBMSUP

The DD statements for the input and output data sets used by DFHBMSUP are:

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 sequential data set or a member of a partitioned data set (DSORG=PO) to contain the BMS macro statements generated by the utility.

Return codes from DFHBMSUP

DFHBMSUP sets one of the following return codes:

0 Utility executed successfully.

4 Input mapset could not be located.

© Copyright IBM Corp. 1989, 2017

8 Output mapset could not be opened.

Example of using DFHBMSUP

Figure 41 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=CICSTS54.CICS.SDFHLOAD,DISP=SHR
//BMSOUT   DD DSN=OUTPUT.MACLIB(MAPOUT),
//         DISP=SHR
//DFHRPL   DD DSN=INPUT.BMSLIB,DISP=SHR
//SYSUDUMP DD SYSOUT=*
//*
```

Figure 41. 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 mapset definition macro, so that you can use it to build HTML template data sets 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'
BMSET40  DFHMSD TYPE=FINAL
         END
```

Chapter 15. Offsite Automatic Reply program (DFH$OFAR)

The offsite automatic reply program (DFH$OFAR) is a NETVIEW executable 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.

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 data sets 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.

DFH$OFAR 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.

The control file should be accessible from any MVS image that runs a participating CICS, and should also be initialized to a list of CICS APPLIDs that are taking part. See “DFH$OFAR control file typical settings” on page 150.

© Copyright IBM Corp. 1989, 2017

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.

DFH$OFAR 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

Return codes from DFH$OFAR

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 16. 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.

These 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. You specify the records which you want to add or remove in control statements read from SYSIN (see "Job control statements to run DFHSMUTL")..

If the local catalog is re-initialized, DFHSMUTL should be run again to add the required subpool records to the local catalog.

Note that deleting and adding a subpool record resets the tuning information for that subpool.

Job control statements to run DFHSMUTL

Control statements are read from SYSIN that specify the storage manager domain subpool records to be added to, or deleted from, the local catalog data set.

The format of the control statements to run DFHSMUTL 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.

```
//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=CICSTS54.CICS.SDFHLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//DFHLCD   DD DSN=CICSTS54.CICS.applid.DFHLCD,DISP=OLD       1
//SYSIN    DD *
ADD SUBPOOL=name-of-subpool-to-add                          2
```

© Copyright IBM Corp. 1989, 2017

```
DEL SUBPOOL=name-of-subpool-to-remove          3
LST
/*
//
```

Note:

1. Change CICSTS54.CICS.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.

Messages from DFHSMUTL

Messages, including errors, are written to SYSPRINT. DFHSM0300 DFHSMUTL REPORT.

1. ADD SUBPOOL=xxxxxxxx PROCESSED SUCCESSFULLY (ADD SUBPOOL=xxxxxxxx has been processed successfully.)
2. DEL SUBPOOL=xxxxxxxx PROCESSED SUCCESSFULLY (DEL SUBPOOL=xxxxxxxx has been processed successfully.)
3. FOUND DFHLCD RECORD SMSUBPOL=xxxxxxxx (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=xxxxxxxx or DEL SUBPOOL=xxxxxxxx are allowed. The statement is ignored.)
7. ERROR PROCESSING 'ADD SUBPOOL=xxxxxxxx'
   .R15 = X'yy'.
   RPL FEEDBACK CODE = X'zz'.
   SEE DFSMS/MVS MACRO INSTRUCTIONS FOR DATA SETS

   (A VSAM error has occurred while processing an ADD SUBPOOL=xxxxxxxx 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=xxxxxxxx'.
   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=xxxxxxxx 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.)

Notices

This information was developed for products and services offered in the U.S.A. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property rights 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 grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America

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

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

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 jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore this statement may not apply to you.

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

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

© Copyright IBM Corp. 1989, 2017

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

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

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this 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.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Programming interface information

CICS supplies some documentation that can be considered to be Programming Interfaces, and some documentation that cannot be considered to be a Programming Interface.

Programming Interfaces that allow the customer to write programs to obtain the services of CICS Transaction Server for z/OS, Version 5 Release 4 are included in the following sections of the online product documentation:

- Developing applications
- Developing system programs
- RACF security overview
- Developing for external interfaces
- Reference: application developmenth
- Reference: system programming
- Reference: connectivity

Information that is NOT intended to be used as a Programming Interface of CICS Transaction Server for z/OS, Version 5 Release 4 , but that might be misconstrued as Programming Interfaces, is included in the following sections of the online product documentation:

- Troubleshooting and support
- Reference: diagnostics

If you access the CICS documentation in manuals in PDF format, Programming Interfaces that allow the customer to write programs to obtain the services of CICS Transaction Server for z/OS, Version 5 Release 4 are included in the following manuals:

- Application Programming Guide and Application Programming Reference
- Business Transaction Services
- Customization Guide
- C++ OO Class Libraries
- Debugging Tools Interfaces Reference
- Distributed Transaction Programming Guide
- External Interfaces Guide
- Front End Programming Interface Guide
- IMS Database Control Guide
- Installation Guide
- Security Guide
- Supplied Transactions
- CICSPlex SM Managing Workloads
- CICSPlex SM Managing Resource Usage
- CICSPlex SM Application Programming Guide and Application Programming Reference
- Java Applications in CICS

If you access the CICS documentation in manuals in PDF format, information that is NOT intended to be used as a Programming Interface of CICS Transaction Server for z/OS, Version 5 Release 4 , but that might be misconstrued as Programming Interfaces, is included in the following manuals:

- Data Areas
- Diagnosis Reference
- Problem Determination Guide
- CICSPlex SM Problem Determination Guide

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at Copyright and trademark information at www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability
These terms and conditions are in addition to any terms of use for the IBM website.

Personal use
You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use
You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM online privacy statement

IBM Software products, including software as a service solutions, (“Software Offerings”) may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering’s use of cookies is set forth below:

For the CICSPlex® SM Web User Interface (main interface):
Depending upon the configurations deployed, this Software Offering may use session and persistent cookies that collect each user’s user name and other personally identifiable information for purposes of session management, authentication, enhanced user usability, or other usage tracking or functional purposes. These cookies cannot be disabled.

For the CICSPlex SM Web User Interface (data interface):
Depending upon the configurations deployed, this Software Offering may use session cookies that collect each user's user name and other personally identifiable information for purposes of session management, authentication, or other usage tracking or functional purposes. These cookies cannot be disabled.

For the CICSPlex SM Web User Interface ("hello world" page):
Depending upon the configurations deployed, this Software Offering may use session cookies that collect no personally identifiable information. These cookies cannot be disabled.

For CICS Explorer®:
Depending upon the configurations deployed, this Software Offering may use session and persistent preferences that collect each user’s user name and password, for purposes of session management, authentication, and single sign-on configuration. These preferences cannot be disabled, although storing a user's password on disk in encrypted form can only be enabled by the user's explicit action to check a check box during sign-on.

If the configurations deployed for this Software Offering provide you, as customer, the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM Privacy Policy and IBM Online Privacy Statement, the section entitled “Cookies, Web Beacons and Other Technologies” and the IBM Software Products and Software-as-a-Service Privacy Statement.

Index

A

auxiliary trace
  using DFHTU to print 53
  using IPCS to print from GTF 61

C

C= keyword, DFHJUP OPTION statement 14
CEMT, master terminal transaction 65
CICS startup
  overriding startup for START=AUTO 137
compiled modules, scanning 107
COND= keyword, DFHJUP OPTION statement 14
CONTROL statement, DFHJUP utility program 12
control statements for DFHDU 66
COPY keyword, DFHJUP OPTION statement 13

D

D= keyword, DFHJUP CONTROL statement 12
D= keyword, DFHJUP OPTION statement 16
data compression
  DFH$MOLS, record expansion 89
DDNAME= keyword, DFHJUP CONTROL statement 12
DDNAME= keyword, DFHJUP OPTION statement 16
DDNOUT= keyword, DFHJUP CONTROL statement 12
DFH$MOLS sample print program 89
  control statements 92, 94
    BREAK 95
    comments 95
    DATE 95
    EXPAND 96
    IGNORE 97
    OPTION 97
    PRINT 98
    rules 94
    SELECT 98
  monitoring sample utility 90
  overview 89
  sample job stream 91
DFH$STED, utility to stagger the end-of-day time 129
DFH$STER
  recovery statistics program 51
  statistics sample program 52
DFH£STXA 37
DFH0STXC 37
DFH0STXD 38
DFH0STXR 36
DFHBMSUP utility program
  introduction 145
DFHCESD, shutdown assist program
  default actions 131
  introduction 131
  sample programs 132
DFHDU transaction dump utility program
  INDEX DD statement for dump summary 66
  job control statements for 69
  processing transaction dump data sets 65
  SYSIN control statements 66
DFHEIDBR 110
DFHEIDTH 110
DFHEISUP (load module scanner) 107
DFHIPCSP, IPCS exit control table member for CICS 72
DFHJUP batch job to read log streams 1
DFHMNDUP dictionary utility program 83
DFHMSCAN utility program, to identify macro-level programs
  detailed report 127
  how the program works 125
  JCL 126
  limitations of 128
  overview 125
  summary report 126
DFHPD, SDUMP formatting exit routine for IPCS 72
  sample jobs for formatting CICS SDUMPs 78
DFHRMUTL, recovery manager utility program
  introduction 137
DFHSMUTL local catalog utility program. 153
DFHSTUP 36
DFHSTUP statistics utility program
  control parameters 42
  job stream for 33
  printing CICS statistics 31
DFHTU trace utility program
  printing auxiliary trace data 53
  selection parameters
    ABBREV 55
    ALL 55
    ENTRY_NUM 55
    EXCEPTION 55
    FULL 55
    KE_NUM 56
    LAST_BLOCKS 56
    PAGESIZE 56
    SHORT 55
    TASKID 56
    TCB_ADDR 56
    TCB_ID 56
    TERMID 56
    TIMERG 57
DFHTU trace utility program *(continued)*
  selection parameters *(continued)*
    TRANID 57
    TYPETR 57
diagnostic run, of CICS 139
dump exit routines 72, 74
  sample jobs for formatting CICS SDUMPs 78
dumps
  CICS transaction dumps 65
  DFHDU utility program, to process CICS transaction dumps 65
  DFHIPCSP, exit control table member for CICS 72
  DFHPD, SDUMP formatting exit routine of IPCS 72
  dump summary 77
    example 78
  error index 77, 78
  exit parameters for CICS 74
  exit routines for IPCS 72, 74
  formatting with IPCS 74
  IPCS exit control table 72
  IPCS verb names for CICS 72
  sample formatting jobs using IPCS 78
  SDUMP options needed for the CICS dump exit 71
  VERBEXIT command of IPCS 74
  via MVS SDUMP macro 65
    processing using IPCS 70
  What to do before formatting with IPCS 71

E

E= keyword, DFHJUP OPTION statement 15
EXITR= keyword, DFHJUP OPTION statement 15
EXTRACT command 39
extract statistics reporting facility 36

F

filter input file for load module scanner 108
FLDLEN= keyword, DFHJUP OPTION statement 14
FLDTYP= keyword, DFHJUP OPTION statement 14

G

generalized trace facility (GTF)
  formatting trace records 62
  GTFTRACE subcommand of IPCS 62
  printing CICS traces using IPCS 61
  sample job to print GTF trace entries 63

© Copyright IBM Corp. 1989, 2017

H

H= keyword, DFHJUP CONTROL statement 12

I

interactive problem control system (IPCS)
- DFHIPCSP, IPCS exit control table member for CICS 72
- dump summary 77, 78
  - example 78
- error index 77, 78
- IPCS verb names for CICS 72
- printing CICS traces from GTF 61
- processing CICS SDUMPs 70
- sample formatting jobs 78
- SDUMP options needed for the CICS dump exit 71

J

journal utility program (DFHJUP)
- control statements
  - COMMENTS 18
  - CONTROL 12
  - END 17
  - OPTION 13
- input and output 9

K

K= keyword, DFHJUP CONTROL statement 12
keywords
- C=, of DFHJUP OPTION statement 14
- COND=, of DFHJUP OPTION statement 14
- COPY, of DFHJUP OPTION statement 13
- D=, of DFHJUP CONTROL statement 12
- D=, of DFHJUP OPTION statement 16
- DDNAME=, of DFHJUP CONTROL statement 12
- DDNAME=, of DFHJUP OPTION statement 16
- DDNOUT=, of DFHJUP CONTROL statement 12
- E=, of DFHJUP OPTION statement 15
- EXITR=, of DFHJUP OPTION statement 15
- FLDLEN=, of DFHJUP OPTION statement 14
- FLDTYP=, of DFHJUP OPTION statement 14
- H=, of DFHJUP CONTROL statement 12
- K=, of DFHJUP CONTROL statement 12
- L=, of DFHJUP OPTION statement 14

keywords *(continued)*
- NEWDCB, of DFHJUP OPTION statement 16
- O=, of DFHJUP CONTROL statement 12
- O=, of DFHJUP OPTION statement 14
- OFFSET=, of DFHJUP OPTION statement 14
- P=, of DFHJUP OPTION statement 16
- PRINT, of DFHJUP OPTION statement 13
- PRTSYS=, of DFHJUP OPTION statement 16
- SKIP=, of DFHJUP CONTROL statement 12
- STOPAFT=, of DFHJUP CONTROL statement 12
- T=, DFHJUP OPTION statement 14
- V=, of DFHJUP OPTION statement 14
- VALUE=, of DFHJUP OPTION statement 14

L

L= keyword, DFHJUP OPTION statement 14
load module scanner
- activating trace 117
- detailed report 114
- summary report 111

load module scanner, utility program 107
- DFHEIDAL, sample filter 110
- DFHEIDBR, sample filter 110
- DFHEIDNT, sample filter 110
- DFHEIDTH, sample filter 110
- exceptions 110
- filter input file 108

local catalog utility program, DFHSMUTL 153
log stream processing
- recommended options for offline processing 8

M

macro-level programs, identifying 125
monitoring
- dictionary utility program, DFHMNDUP 83
- sample print program, DFH$MOLS 89

N

NEGOF keyword, DFHJUP OPTION statement 14
NEWDCB, DFHJUP OPTION statement 16

O

O= keyword, DFHJUP CONTROL statement 12
O= keyword, DFHJUP OPTION statement 14
OFFSET= keyword, DFHJUP OPTION statement 14
Offsite automatic reply program 149

P

P= keyword, DFHJUP OPTION statement 16
PRINT keyword, DFHJUP OPTION statement 13
PRTSYS= keyword, DFHJUP OPTION statement 16

R

recovery manager domain 51
recovery manager utility program, DFHRMUTL
- introduction 137

recovery messages
- Recovery Manager domain 51

recovery statistics program, DFH$STER 51

S

SDTRAN, system initialization parameter 131
shutdown assist program, DFHCESD
- default actions 131
- introduction 131
- sample programs 132

SKIP= keyword, DFHJUP CONTROL statement 12
staggering the end-of-day time 129
START, system initialization parameter 137
statistics sample program, DFH$STER 52
statistics sample programs
- DFH£STXA 37
- DFH0STXC 37
- DFH0STXD 38
- DFH0STXR 36

statistics utility program, DFHSTUP 31
statistics, staggering the end-of-day time 129
STOPAFT= keyword, DFHJUP CONTROL statement 12
SUBSYS= keyword 1
system initialization parameters
- SDTRAN 131
- START 137

T

T= keyword, DFHJUP OPTION statement 14
trace
- using DFHTU to print 53

trace *(continued)*
using IPCS to print from GTF 61
utility programs 53
trace utility program, DFHTU 53

U

utility program, to scan for API commands
overview 107
utility programs, offline
assisting in disaster recovery of a CICSplex, DFH$OFAR 149
batch-enabling sample programs for RLS access-mode data sets 119
default shutdown assist program, DFHCESD 131
identifying macro-level programs (DFHMSCAN) 125
load module scanner, scanning for API commands 107
preparing statistics reports, DFHSTUP 31
preparing trace reports, DFHTU 53
processing CICS monitoring data, DFH$MOLS 89
processing CICS monitoring data, DFHMNDUP 83
processing log data, DFHJUP 1
processing the global catalog data set, DFHRMUTL 137
processing transaction dump data sets (DFHDU) 65
recreating BMS macro statements, DFHBMSUP 145
staggering the end-of-day time (DFH$STED) 129

V

V= keyword, DFHJUP OPTION statement 14
VALUE= keyword, DFHJUP OPTION statement 14
VERBEXIT subcommand of IPCS 70, 74
exit parameters for CICS 74

